Semantic Web 2 (2011) 125–155
DOI 10.3233/SW-2011-0041
IOS Press

Is Question Answering fit for the Semantic Web?: A survey

Editor(s): Philipp Cimiano, Universität Bielefeld, Germany
Solicited review(s): three anonymous reviewers

Vanessa Lopez[a,*], Victoria Uren[b], Marta Sabou[c] and Enrico Motta[a]
[a] *Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, MK7 6AA, United Kingdom*
[b] *The University of Sheffield, S14DP, United Kingdom*
[c] *MODUL University of Vienna, Austria*

Abstract. With the recent rapid growth of the Semantic Web (SW), the processes of searching and querying content that is both massive in scale and heterogeneous have become increasingly challenging. User-friendly interfaces, which can support end users in querying and exploring this novel and diverse, structured information space, are needed to make the vision of the SW a reality. We present a survey on ontology-based Question Answering (QA), which has emerged in recent years to exploit the opportunities offered by structured semantic information on the Web. First, we provide a comprehensive perspective by analyzing the general background and history of the QA research field, from influential works from the artificial intelligence and database communities developed in the 70s and later decades, through open domain QA stimulated by the QA track in TREC since 1999, to the latest commercial semantic QA solutions, before tacking the current state of the art in open user-friendly interfaces for the SW. Second, we examine the potential of this technology to go beyond the current state of the art to support end-users in reusing and querying the SW content. We conclude our review with an outlook for this novel research area, focusing in particular on the R&D directions that need to be pursued to realize the goal of efficient and competent retrieval and integration of answers from large scale, heterogeneous, and continuously evolving semantic sources.

Keywords: Question Answering survey, Natural Language, Semantic Web, ontology

1. Introduction

The emerging Semantic Web (SW) [7] offers a wealth of semantic data about a wide range of topics, representing real community agreement. We are quickly reaching the critical mass required to enable a true vision of a large scale, distributed SW with real-world datasets, leading to new research possibilities that can benefit from exploiting and reusing this vast resources, unprecedented in the history of computer science. Hence, there is now a renewed interest in the search engine market towards the introduction of semantics in order to improve over current keyword search technologies [6,29,43].

The notion of introducing semantics to search on the Web is not understood in a unique way. According to [29] the two most common uses of SW technology are: (1) to interpret Web queries and Web resources annotated with respect to the background knowledge described by underlying ontologies, and (2) to search in the structured large datasets and Knowledge Bases (KBs) of the SW as an alternative or a complement to the current web.

Apart from the benefits that can be obtained as more semantic data is published on the Web, the emergence and continued growth of a large scale SW poses some challenges and drawbacks:

– There is a gap between users and the SW: it is difficult for end-users to understand the com-

[*] Corresponding author. E-mail: v.lopez@open.ac.uk.

plexity of the logic-based SW. Solutions that can allow the typical Web user to profit from the expressive power of SW data-models, while hiding the complexity behind them, are of crucial importance.

- The processes of searching and querying content that is massive in scale and highly heterogeneous have become increasingly challenging: current approaches to querying semantic data have difficulties to scale their models successfully to cope with the increasing amount of distributed semantic data available online.

Hence, there is a need for user-friendly interfaces that can scale up to the Web of Data, to support end users in querying this heterogeneous information space.

Consistent with the role played by ontologies in structuring semantic information on the Web, recent years have witnessed the rise of ontology-based Question Answering (QA) as a new paradigm of research, to exploit the expressive power of ontologies and go beyond the relatively impoverished representation of user information needs in keyword-based queries. QA systems have been investigated by several communities [45], e.g., Information Retrieval (IR), artificial intelligence and database communities. Traditionally, QA approaches have largely been focused on retrieving answers from raw text, with the emphasis on using ontologies to mark-up Web resources and improve retrieval by using query expansion [70]. The novelty of this trend of ontology-based QA is to exploit the SW information for making sense of, and answering, user queries.

In this paper, we present a survey of ontology-based QA systems and other related work. We look at the promises of this novel research area from two perspectives. First, its contributions to the area of QA systems in general; and second, its potential to go beyond the current state of the art in SW interfaces for end-users, thus, helping to bridge the gap between the user and the SW.

We seek a comprehensive perspective on this novel area by analyzing the key dimensions in the formulations of the QA problem in Section 2. We classify a QA system, or any approach to query the SW content, according to four dimensions based on the type of questions (input), the sources (unstructured data such as documents, or structured data in a semantic or non-semantic space), the scope (domain-specific, open-domain), and the traditional intrinsic problems derived from the search environment and scope of the system. To start with, we introduce in Section 3 the general background and history of the QA research field, from the influential works in the early days of research on architectures for Natural Language Interfaces to Databases (NLIDB) in the 70s (Section 3.1), through the approaches to open domain QA over text (Section 3.2), to the latest proprietary (commercial) semantic QA systems, based on data that is by and large manually coded and homogeneous (Section 3.3). Then, in Section 4 we discuss the state of the art in ontology-based QA systems (Section 4.1), in particular analyzing their drawbacks (restricted domain) when considering the SW in the large (Section 4.2). We then review the latest trends in open domain QA interfaces for the SW (Section 4.3) and look at the evaluations that have been conducted to test them (Section 4.4). We finish this section with a discussion on the competences of these systems in the QA scenario (Section 4.5), highlighting the open issues (Section 4.6). In Section 5, we focus on approaches developed in the last decade, that have attempted to support end users in querying the SW data in the large, from early global-view information systems (Section 5.1) and restricted domain semantic search (Section 5.2), to the latest works on open domain large scale semantic search and Linked Data [9] interfaces (Section 5.3). In Section 6, we argue that this new ontology-based search paradigm based on natural language QA, is a promising direction towards the realization of user-friendly interfaces for all the analyzed dimensions, as it allows users to express arbitrarily complex information needs in an intuitive fashion. We conclude in Section 7 with an outlook for this research area, in particular, our view on the potential directions ahead to realize its ultimate goal: to retrieve and combine answers from multiple, heterogeneous and automatically discovered semantic sources.

2. Goals and dimensions of Question Answering

The goal of QA systems, as defined by [45], is to allow users to ask questions in Natural Language (NL), using their own terminology, and receive a concise answer. In this section, we give an overview of the multiple dimensions in the QA process. These dimensions can be extended beyond NL QA systems to any approach to help users to locate and query structured data on the Web.

We can classify a QA system, and any semantic approach for searching and querying SW content, according to four interlinked dimensions (see Fig. 1): (1) the input or type of questions it is able to accept (facts, dialogs, etc); (2) the sources from which it can

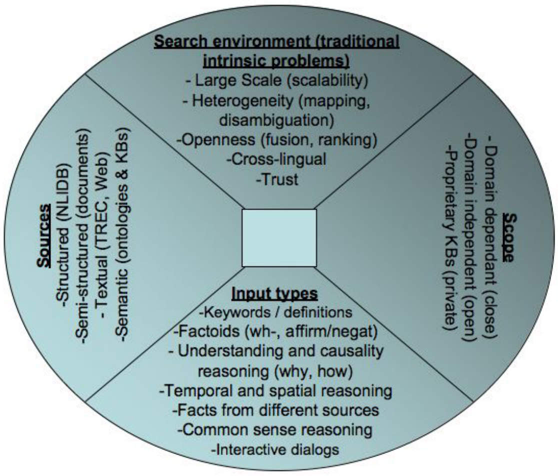
Search environment (traditional intrinsic problems)
- Large Scale (scalability)
- Heterogeneity (mapping, disambiguation)
-Openness (fusion, ranking)
-Cross-lingual
-Trust
Sources
-Structured (NLIDB)
-Semi-structured (documents)
- Textual (TREC, Web)
-Semantic (ontologies & KBs)
Scope
- Domain dependant (close)
-Domain independent (open)
-Proprietary KBs (private)
Input types
-Keywords / definitions
-Factoids (wh-, affirm/negat)
- Understanding and causality reasoning (why, how)
-Temporal and spatial reasoning
-Facts from different sources
-Common sense reasoning
-Interactive dialogs

Fig. 1. The dimensions of Question Answering and query and search interfaces in general.

derive the answers (structured vs. unstructured data); (3) the scope (domain specific vs. domain independent), and (4) how it copes with the traditional intrinsic problems that the search environment imposes in any non-trivial search system (e.g., adaptability and ambiguity).

At the **input** level, the issue is balancing usability and higher expressivity at the level of the query, hiding the complexity of SQL-like query languages, while allowing the user to express his / her information needs fully. Different kinds of search inputs provide complementary affordances to support the ordinary user in querying the semantic data. The best feature of keyword-based search is its simplicity. Nevertheless, in this simplicity lie its main limitations: the lack of expressivity, e.g., in expressing relationships between words, and the lack of context to disambiguate between different interpretations of the keywords. In [79], QA systems are classified, according to the complexity of the input question and the difficulty of extracting the answer, in five increasingly sophisticated types: systems capable of processing factual questions (factoids), systems enabling reasoning mechanisms, systems that fuse answers from different sources, interactive (dialog) systems and systems capable of analogical reasoning. Most research in QA focuses on factual QA, where we can distinguish between *Wh-queries* (who, what, how many, etc.), commands (name all, give me, etc.) requiring an element or list of elements as an answer, or affirmation / negation questions. As pointed out in [47] more difficult kinds of factual questions include those which ask for opinion, like *Why* or *How* questions, which require understanding of causality or instrumental relations, *What* questions which provide little constraint in the answer type, and definition questions. In this survey we focus on factual QA, including open-domain definition questions, i.e., *What*-queries about arbitrary concepts. In the SW context factual QA means that answers are ground facts as typically found in KBs and provides an initial foundation to tackle more ambitious forms of QA.

QA systems can also be classified according to the different **sources** used to generate an answer as follows:

- Natural Language interfaces to structured data on databases (NLIDB traced back to the late sixties [3]).
- QA over semi-structured data (e.g., health records, yellow pages, wikipedia infoboxes).
- Open QA over free text, fostered by the open-domain QA track introduced by TREC (http://trec.nist.gov) in 1999 (TREC-8).
- QA over structured semantic data, where the semantics contained in ontologies provide the context needed to solve ambiguities, interpret and answer the user query.

Another distinction between QA systems is whether they are domain-specific (closed domain) or domain-independent (open domain). Ontology-based QA emerged as a combination of ideas of two different research areas – it enhances the **scope** of closed NLIDB over structured data, by being agnostic to the domain of the ontology that it exploits; and also presents complementary affordances to open QA over free text (TREC), the advantage being that it can help with answering questions requiring situation-specific answers, where multiple pieces of information (from one or several sources) need to be assembled to infer the answers at run time. Nonetheless, most ontology-based QA systems are akin to NLIDB in the sense that they are able to extract precise answers from structured data in a specific domain scenario, instead of retrieving relevant paragraphs of text in an open scenario. Latest proprietary QA systems over structured data, such as TrueKnowledge and Powerset (detailed in Section 3.3), are open domain but restricted to their own proprietary sources.

A challenge for domain-independent systems comes from the **search environment** that can be characterized by large scale, heterogeneity, openness and multilinguality. The search environment influences to what level semantic systems perform a deep exploitation of the semantic data. In order to take full advantage of the inherent characteristics of the semantic information space to extract the most accurate answers for the users, QA systems need to tackle various **traditional intrinsic** problems derived from the search environment, such as:

- Mapping the terminology and information needs of the user into the terminology used by the sources, in such a form that: (1) it can be evaluated using standard query processing and inferencing techniques, (2) it does not affect portability or adaptability of the systems to new domains, and (3) it leads to the correct answer.
- Disambiguating between all possible interpretations of a user query. Independently of the type of query, any non-trivial NL QA system has to deal with ambiguity. Furthermore, in an open scenario, ambiguity cannot be solved by means of an internal unambiguous knowledge representation, as in domain-restricted scenarios. In open-domain scenarios, systems face the problem of polysemous words, with different meanings according to different domains.
- Because answers may come from different sources, and different sources have varying levels of quality and trust, knowledge fusion and ranking measures should be applied to select the better sources, fuse similar answers together, and rank the answers across sources.
- With regards to scalability, in general terms, there is a trade-off between the complexity of the querying process and the amount of data systems can use in response to a user demand in a reasonable time.

Multilinguality issues, the ability to answer a question posed in one language using an answer corpus in another language, fostered by the Multilingual Question Answering Track at the cross language evaluation forum (CLEF)[1] since 2002 [33], are not reviewed in this survey. This is because in the context of QA in the open SW, challenges such as scalability and heterogeneity need to be tackled first to obtain answers across sources.

NL interfaces are an often-proposed solution in the literature for casual users [51], being particularly appropriate in domains for which there are authoritative and comprehensive databases or resources [80]. However, their success has been typically overshadowed by both the brittleness and *habitability* problems [91], defined as the mismatch between the user expectations and the capabilities of the system with respect to its NL understanding and what it knows about (users do not know what it is possible to ask). As stated in [94] iterative and exploratory search modes are important to the **usability** of all search systems, to support the user in understanding what is the knowledge of the system and what subset of NL is possible to ask about. Systems also should be able to provide justifications for an answer in an intuitive way (NL generation), suggest the presence of unrequested but related information, and actively help the user by recommending searches or proposing alternate paths of exploration. For example, view based search and forms can help the user to explore the search space better than keyword-based or NL querying systems, but they become tedious to use in large spaces and impossible in heterogeneous ones.

Usability of NL interfaces is not covered in this review so for additional information we refer the reader to [94] and [51].

3. Related work on Question Answering

Here we present a short survey of related work on QA targeted to different types of sources: structured databases, unstructured free text and precompiled semantic KBs.

3.1. NLIDB: Natural Language interfaces to databases

The use of NL to access relational databases can be traced back to the late sixties and early seventies [3]. The first QA systems were developed in the sixties and they were basically NL interfaces to expert systems, tailored to specific domains, the most famous ones being **BASEBALL** [35] and **LUNAR** [98] Both systems were domain specific, the former answered questions about the US baseball league over the period of one year, the later answered questions about the geological analysis of rocks returned by the Apollo missions. LUNAR was able to answer 90% of the questions in its domain when posed by untrained geologists. In [3] a detailed overview of the state of the art for these early systems can be found.

Some of the early NLIDB approaches relied on pattern-matching techniques. In the example described by [3], a rule says that if a user's request contains the word "capital" followed by a country name, the system should print the capital which corresponds

[1] http://clef.isti.cnr.it

to the country name, so the same rule will handle "what is the capital of Italy?", "print the capital of Italy", "could you please tell me the capital of Italy". This shallowness of the pattern-matching would often lead to failures but it has also been an unexpectedly effective technique for exploiting domain-specific data sources.

The main drawback of these early NLIDB systems is that they were built having a particular database in mind, thus they could not be easily modified to be used with different databases and were difficult to port to different application domains. Configuration phases were tedious and required a long time, because of domain-specific grammars, hard-wired knowledge or hand-written mapping rules that had to be developed by domain experts.

The next generation of NLIDBs used an intermediate representation language, which expressed the meaning of the user's question in terms of high-level concepts, independently of the database's structure [3]. Thus, separating the (domain-independent) linguistic process from the (domain-dependent) mapping process into the database, to improve the portability of the front end [68].

The **formal semantics approach** presented in [25] follows this paradigm and clearly separates between the NL front ends, which have a very high degree of portability, from the back end. The front end provides a mapping between sentences of English and expressions of a formal semantic theory, and the back end maps these into expressions, which are meaningful with respect to the domain in question. Adapting a developed system to a new application requires altering the domain specific back end alone.

MASQUE/SQL [2] is a portable NL front end to SQL databases. It first translates the NL query into an intermediate logic representation, and then translates the logic query into SQL. The semi-automatic configuration procedure uses a built-in domain editor, which helps the user to describe the entity types to which the database refers, using an is-a hierarchy, and then to declare the words expected to appear in the NL questions and to define their meaning in terms of a logic predicate that is linked to a database table/view.

More recent work in the area (2003) can be found in **PRECISE** [85]. PRECISE maps questions to the corresponding SQL query by identifying classes of questions that are understood in a well defined sense: the paper defines a formal notion of *semantically tractable* questions. Questions are translated into sets of attribute/value pairs and a relation token corresponds to either an attribute token or a value token. Each attribute in the database is associated with a wh-value (what, where, etc.). Also, a lexicon is used to find synonyms. The database elements selected by the matcher are assembled into a SQL query, if more than one possible query is found, the user is asked to choose between the possible interpretations. However, in PRECISE the problem of finding a mapping from the tokenization to the database requires all tokens to be distinct; questions with unknown words are not semantically tractable and cannot be handled. As a consequence, PRECISE will not answer a question that contains words absent from its lexicon. Using the example suggested in [85], the question "what are some of the neighbourhoods of Chicago?" cannot be handled by PRECISE because the word "neighbourhood" is unknown. When tested on several hundred questions, 80% of them were semantically tractable questions, which PRECISE answered correctly, and the other 20% were not handled.

NLI have attracted considerable interest in the Health Care area. In the approach presented in [41] users can pose complex NL queries to a large medical repository, question formulation is facilitated by means of ***Conceptual Authoring***. A logical representation is constructed using a query editing NL interface, where, instead of typing in text, all editing operations are defined directly on an underlying logical representation governed by a predefined ontology ensuring that no problem of interpretation arises.

However, all these approaches still need an intensive configuration procedure. To reduce the formal complexity of creating underlying grammars for different domains, [75], and most recently **C-PHRASE** [74] present a state-of-the-art authoring system for NLIDB. The author builds the semantic grammar through a series of naming, tailoring and defining operations within a web-based GUI, as such the NLI can be configured by non-specialized, web based technical teams. In that system queries are represented as expressions in an extended version of Codd's Tuple Calculus, which may be directly mapped to SQL queries or first-order logic expressions. Higher-order predicates are also used to support ranking and superlatives.

3.2. Open domain Question Answering over text

3.2.1. Document-based Question Answering

Most current work on QA, which has been rekindled largely by the TREC Text Retrieval Conference (sponsored by the American National Institute, NIST, and the Defense Advanced Research Projects Agency, DARPA) and by the cross-lingual QA Track at CLEF, is somewhat different in nature from querying structured data. These campaigns enable research in QA from the IR perspective, where the task consists in finding the text that contains the answer to the question and extracting the answer. The ARDA's Advanced Question Answering for Intelligence funded the AQUAINT program, a multi-project effort to improve the performance of QA systems over free large heterogeneous collections of structured and unstructured text or media. Given the large, uncontrolled text files and the very weak world knowledge available from WordNet and gazetteers, these systems have performed surprisingly well. For example, the **LCC system** [77] that uses a deep linguistic analysis and iterative strategy obtained a score of 0.856 by answering correctly 415 questions out of 500 in TREC-11 (2002).

There are linguistic problems common in most kinds of NL understanding systems. A high-level overview on the state of the art techniques for open QA can be found in [83]. Some of the methods use shallow keyword-based expansion techniques to locate interesting sentences from the retrieved documents, based on the presence of words that refer to entities of the same type of the expected answer type. Ranking is based on syntactic features such as word order or similarity to the query. Templates can be used to find answers that are just reformulations of the question. Most of the systems classify the query based on the type of the answer expected: e.g., a name (i.e., person, organization), a quantity (monetary value, distance, length, size) or a date. Classes of questions are arranged hierarchically in taxonomies and different types of questions require different strategies. These systems often utilize world knowledge that can be found in large lexical resources such as WordNet, or ontologies such as Suggested Upper Merged Ontology (SUMO) to pinpoint question types and match entities to the expected answer type. More sophisticated syntactic, semantic and contextual processing to construct an answer might include: named-entity (NE) recognition, relation extraction, co-reference resolution, syntactic alternations, word sense disambiguation (WSD), logical inferences and temporal-spatial reasoning.

Going into more details, QA applications for text typically involve two steps, as pointed out by [45]: (1) "identifying the semantic type of the entity sought by the question"; and (2) "determining additional constraints on the answer entity". Constraints can include, for example, keywords (that may be expanded using synonyms or morphological variants) to be used in the matching of candidate answers, and syntactic or semantic relations between a candidate answer entity and other entities in the question. Various systems have, therefore built hierarchies of question types based on the types of answers sought [46,78,87,99]. NE recognition and information extraction (IE) are powerful tools in free text QA. The study presented in [87] showed that over 80% of questions asked for a named entity as a response.

For instance, in **LASSO** [78] a question type hierarchy was constructed from the analysis of the TREC-8 training data, and a score of 55.5% for short answers and 64.5% for long answers was achieved. Given a question, LASSO can find automatically (a) the type of the question (what, why, who, how, where), (b) the type of the answer (person, location, etc.), (c) the focus of the question, defined as the "main information required by the interrogation" (useful for "what" questions, which usually leave implicit the type of the answer which is sought), (d) the relevant keywords from the question. Occasionally, some words of the question do not occur in the answer (for example, the focus "day of the week" is very unlikely to appear in the answer). Therefore, LASSO implements NE recognition heuristics for locating the possible answers.

The best results of the TREC-9 competition were obtained by the **FALCON** system described in [42], with a score of 58% for short answers and 76% for long answers. In FALCON the semantic categories of the answers are mapped into categories covered by a NE Recognizer. When the answer type is identified, it is mapped into an answer taxonomy, where the top categories are connected to several word classes from WordNet. In an example presented in [42], FALCON identifies the expected answer type of the question "what do penguins eat?" as food because "it is the most widely used concept in the glosses of the subhierarchy of the noun synset {eating, feeding}". All nouns (and lexical alterations), immediately related to the concept that determines the answer type, are considered among the other query keywords. Also, FALCON gives a cached answer if the similar question has already been asked before; a similarity measure is calculated to see if the given question is a reformulation of a previous one.

The system described in Litkowski et al. [61], called **DIMAP**, extracts "semantic relation triples" after a document is parsed, converting a document into triples. The DIMAP triples are stored in a database in order to be used to answer the question. The semantic relation triple described consists of a discourse entity, a semantic relation that characterizes the entity's role in the sentence and a governing word to which the entity stands in the semantic relation. The parsing process generates an average of 9.8 triples per sentence in a document. The same analysis was done for each question, generating on average 3.3 triples per sentence, with one triple for each question containing an unbound variable, corresponding to the type of question (the system categorized questions in six types: time, location, who, what, size and number questions).

3.2.2. Question Answering on the Web

QA systems over the Web have the same three main components as QA systems designed to extract answers to factual questions by consulting a repository of documents (TREC): (1) a query formulation mechanism that translates the NL queries into the required IR queries, (2) a search engine over the Web, instead of an IR engine searching the documents, and (3) the answer extraction module that extracts answers from the retrieved documents. A technique commonly shared in Web and TREC-systems, is to use WordNet or NE tagging to classify the type of the answer.

For instance, **Mulder** [55] is a QA system for factual questions over the Web, which relies on multiple queries sent to the search engine Google. To form the right queries for the search engine, the query is classified using WordNet to determine the type of the object of the verb in the question (numerical, nominal, temporal), then a reformulation module converts a question into a set of keyword queries by using different strategies: extracting the most important keywords, quoting partial sentences (detecting noun phrases), conjugating the verb, or performing query expansion with WordNet. In Mulder, an answer is extracted from the snippets or summaries returned by Google, which is less expensive than extracting answers directly from a Web page. Then, to reduce the noise or incorrect information typically found on the Web and improve accuracy, Mulder clusters similar answers together and picks the best answer with a voting procedure. Mulder takes advantage of Google ranking algorithms base on PageRank, the proximity or frequency of the words, and the wider coverage provided by Google: "with a large collection there is a higher probability of finding target sentences". An evaluation using the TREC-8 questions, based on the Web, instead of the TREC document collection, showed that Mulder's recall is more than a factor of three higher than AskJeeves.

The search engine **AskJeeves**[2] looks up the user's question in its database and returns a list of matching questions that it knows how to answer, the user selects the most appropriate entry in the list, and he is taken to the Web pages where the answer can be found. AskJeeves relies on human editors to match question templates with authoritative sites.

Other approaches are based on statistical or semantic similarities. For example, **FAQ Finder** [12] is a NL QA system that uses files of FAQs as its KB; it uses two metrics to match questions to answers: statistical similarity and semantic similarity. For shorter answers over limited structured data, NLP-based systems have generally performed better than statistical based ones, which need a lot of domain specific training and long documents with large quantities of data containing enough words for statistical comparisons to be considered meaningful. Semantic similarity scores rely on finding connections through WordNet between the user's question and the answer. The main problem here is the inability to cope with words that are not explicitly found in the KB. **Gurevych's** [40] **approach** tries to identify semantically equivalent questions, which are paraphrases of user queries, already answered in social Q&A sites, such as Yahoo!Answers.

Finally, Google itself is also evolving into a NL search engine, providing precise answers to some specific factual queries, together with the Web pages from which the answers have been obtained. However, it does not yet distinguish between queries such as "where Barack Obama was born" or "when Barack Obama was born" (as per May 2011).

3.3. Latest developments on structured (proprietary) open Question Answering

As we have seen in the previous subsections, large-scale, open-domain QA has been stimulated in the last decade (since 1999) by the TREC QA track evaluations. The current trend is to introduce semantics to search for Web pages based on the meaning of the words in the query, rather than just matching keywords and ranking pages by popularity. Within this context, there are also approaches that focus on directly obtaining structured answers to user queries from pre-compiled semantic information, which is used to understand and disambiguate the intended meaning and relationships of the words in the query.

[2] http://www.ask.co.uk

This class of systems includes START, which came online in 1993 as the first QA system available on the Web, and several industrial startups such as Powerset, Wolfram Alpha and True Knowledge[3], among others. These systems use a well-established approach, which consists of semi-automatically building their own homogeneous, comprehensive factual KB about the world, similarly to OpenCyc and Freebase[4].

START [49] answers questions about geography and the MIT infolab, with a performance of 67% over 326 thousand queries. It uses highly edited KBs to retrieve tuples in the subject-relation-object form, as pointed out by [49], although not all possible queries can be represented in the binary relational model, in practice these exceptions occur very infrequently. START compares the user query against the annotations derived from the KB. However, START suffers from the knowledge acquisition bottleneck, as only trained individuals can add knowledge and expand the system's coverage (by integrating new Web sources).

Commercial systems include PowerSet, which tries to match the meaning of a query with the meaning of a sentence in Wikipedia. Powerset not only works on the query side of the search (converting the NL queries into database understandable queries, and then highlighting the relevant passage of the document), but it also reads every word of every (Wikipedia) page to extract the semantic meaning. It does so by compiling *factzs* – similar to triples, from pages across Wikipedia, together with the Wikipedia page locations and sentences that support each factz and using Freebase and its semantic resources to annotate them. The Wolfram Alpha knowledge inference engine builds a broad trusted KB about the world by ingesting massive amounts of information (approx. 10TBs, still a tiny fraction of the Web), while True Knowledge relies on users to add and curate information.

4. Semantic ontology-based Question Answering

In this section we look at ontology-based semantic QA systems (also referred in this paper as semantic QA systems), which take queries expressed in NL and a given ontology as input, and return answers drawn from one or more KBs that subscribe to the ontology. Therefore, they do not require the user to learn the vocabulary or structure of the ontology to be queried.

4.1. Ontology-specific QA systems

Since the steady growth of the SW and the emergence of large-scale semantics the necessity of NLI to ontology-based repositories has become more acute, re-igniting interest in NL front ends. This trend has also been supported by usability studies [51], which show that casual users, typically overwhelmed by the formal logic of the SW, prefer to use a NL interface to query an ontology. Hence, in the past few years there has been much interest in ontology based QA systems, where the power of ontologies as a model of knowledge is directly exploited for the query analysis and translation, thus providing a new twist on the old issues of NLIDB, by focusing on portability and performance, and replacing the costly domain specific NLP techniques with shallow but domain-independent ones. A wide range of off-the-shelf components, including triple stores (e.g., Sesame[5]) or text retrieval engines (e.g., Lucene[6]), domain-independent linguistic resources, such as WordNet and FrameNet[7], and NLP Parsers, such as Stanford Parser (Klein and Manning, 2002), support the evolution of these new NLI.

Ontology-based QA systems vary on two main aspects: (1) the degree of domain customization they require, which correlates with their retrieval performance, and (2) the subset of NL they are able to understand (full grammar-based NL, controlled or guided NL, pattern based), in order to reduce both complexity and the habitability problem, pointed out as the main issue that hampers the successful use of NLI [51].

At one end of the spectrum, systems are tailored to a domain and most of the customization has to be performed or supervised by domain experts. For instance QACID [32] is based on a collection of queries from a given domain that are analyzed and grouped into clusters, where each cluster, containing alternative formulations of the same query, is manually associated with SPARQL queries. In the mid-

[3] http://www.powerset.com/, http://www.wolframalpha.com/index.html, and http://www.trueknowledge.com/

[4] www.opencyc.org, http://www.freebase.com

[5] http://www.openrdf.org/

[6] http://lucene.apache.org/

[7] http://wordnet.princeton.edu, http://framenet.icsi.berkeley.edu

Table 1

Ontology-based QA approaches classified by the subset of NL and degree of customization

Ontology-based QA systems	Subset of NL			Customization		Ontology-independent		
	Guided NL	Bag of words	Full shallow grammar	Domain grammar / collection	Domain lexicons	User learning	Relation (Triple) based	Pattern-matching (structural lexicon)
QACID		+		+	+			
ORAKEL			+	+	+			
e-Librarian			+		+			
GINSENG	+							+
NLPReduce		+						+
Querix			+				+	+
AquaLog			+			+	+	(entity lexicon only)
PANTO			+				+	+
QuestIO		+					+	+ (gazetteers)
FreyA			+			+	+	+ (gazetters)

dle of the spectrum, a system such as ORAKEL [15] requires a significant domain-specific lexicon customization process, while for systems like the e-librarian [59] performance is dependent on the manual creation of a domain dependent lexicon and dictionary. At the other end of the spectrum, in systems like AquaLog [67], the customization is done on the fly while the system is being used, by using interactivity to learn the jargon of the user over time. GINSENG [8] guides the user through menus to specify NL queries, while systems such as PANTO [96], NLP-Reduce, Querix [53] and QuestIO [88], generate lexicons, or ontology annotations (FREya by Damljanovic et al.), on demand when a KB is loaded. In what follows, we look into these systems in detail and present a comparison in Table 1.

AquaLog [67] allows the user to choose an ontology and then ask NL queries with respect to the universe of discourse covered by the ontology. AquaLog is ontology independent because the configuration time required to customize the system for a particular ontology is negligible. The reason for this is that the architecture of the system and the reasoning methods are completely domain-independent, relying on the semantics of the ontology, and the use of generic lexical resources, such as WordNet. In a first step, the Linguistic Component uses the GATE infrastructure and resources [20] to obtain a set of linguistic annotations associated with the input query. The set of annotations is extended by the use of JAPE grammars[8] to identify terms, relations, question indicators (who, what, etc.), features (voice and tense) and to classify the query into a category. Knowing the category and GATE annotations for the query, the Linguistic Component creates the linguistic triples or Query-Triples. Then, these Query-Triples are further processed and interpreted by the Relation Similarity Service, which maps the Query-Triples to ontology-compliant Onto-Triples, from which an answer is derived. AquaLog identifies ontology mappings for all the terms and relations in the Query-Triples by means of string based comparison methods and WordNet. In addition, AquaLog's interactive relation similarity service uses the ontology taxonomy and relationships to disambiguate between the alternative representations of the user query. When the ambiguity cannot be resolved by domain knowledge the user is asked to choose between the alternative readings. AquaLog includes a learning component to automatically obtain domain-dependent knowledge by creating a lexicon, which ensures that the performance of the system improves over time, in response to the particular community jargon (vocabulary) used by end users. AquaLog uses generalization rules to learn novel associations between the NL relations used by the users and the ontology structure. Once the question is entirely mapped to the underlying ontological structure the corresponding instances are obtained as an answer.

QACID [32] relies on the ontology, a collection of user queries, and an entailment engine that associates new queries to a cluster of existing queries. Each query is considered as a bag of words, the mapping between words in NL queries to instances in a KB is done through string distance metrics [17] and an

[8] JAPE is a language for creating regular expressions applied to linguistic annotations in a text corpus

ontological lexicon. Prior to launching the corresponding SPARQL query for the cluster, the SPARQL generator replaces the ontology concepts with the instances mapped for the original NL query. This system is at the domain-specific end of the spectrum because the performance depends on the variety of questions collected in the domain, the process is domain-dependent, costly and can only be applied to domains with limited coverage.

ORAKEL [15] is a NL interface that translates factual *wh-queries* into F-logic or SPARQL and evaluates them with respect to a given KB. The main feature is that it makes use of a compositional semantic construction approach thus being able to handle questions involving quantification, conjunction and negation. In order to translate factual wh-queries it uses an underlying syntactic theory built on a variant of a *Lexicalized Tree Adjoining Grammar* (LTAG), extended to include ontological information. The parser makes use of two different lexicons: the general lexicon and the domain lexicon. The general or domain independent lexicon includes closed-class words such as determiners, i.e., a, the, every, etc., as well as question pronouns, i.e., who, which, etc. The domain lexicon, in which natural expressions, verbs, adjectives and relational nouns, are mapped to corresponding relations specified in the domain ontology, varies from application to application and, for each application, this lexicon has to be partially generated by a domain expert. The semantic representation of the words in the domain independent lexicon makes reference to domain independent categories, as given for example by a foundational ontology such as DOLCE. This assumes that the domain ontology is somehow aligned to the foundational categories provided by the foundational ontology. Therefore, the domain expert is only involved in the creation of the domain specific lexicon, which is actually the most important lexicon as it is the one containing the mapping of linguistic expressions to domain-specific predicates. The domain expert has to instantiate *subcategorization frames*, which represent linguistic structures (e.g., verbs with their arguments), and maps these to domain-specific relations in the ontology. WordNet is used with the purpose to suggest synonyms (in the most frequent sense of the word) for the verb or noun currently edited. The approach is independent of the target language, which only requires a declarative description in Prolog of the transformation from the logical form to the target language.

The **e-Librarian** [59] understands the sense of the user query to retrieve multimedia resources from a KB. First, the NL query is pre-processed into its linguistic classes, in the form of triples, and translated into an unambiguous logical form, by mapping the query to an ontology to solve ambiguities. If a query is composed of several linguistic clauses, each one is translated separately and the logical concatenation depends on the conjunction words used in the question. The system relies on simple, string-based comparison methods (e.g., *edit distance metrics*) and a domain dictionary to look up lexically related words (synonyms) because general-purpose dictionaries like WordNet are often not appropriate for specific domains. Regarding portability, the creation of this dictionary is costly, as it has to be created for each domain, but the strong advantage of this is that it provides very high performance, which is difficult to obtain with general-purpose dictionaries (from 229 user queries, 97% were correctly answered in the evaluation). The e-librarian does not return the answer to the user's question, but it retrieves the most pertinent document(s) in which the user finds the answer to her question.

Moving into the systems that do not necessitate any customization effort or previous pre-processing, [51] presented four different ontology-independent query interfaces with the purpose of studying the usability of NLI for casual end-users. These four systems lie at different positions of what they call the *Formality Continuum*, where the freedom of a full NL and the structuredness of a formal query language are at opposite ends of the continuum. The first two interfaces, *NLP-Reduce* and *Querix* allow users to pose questions in full or slightly controlled English. The third interface *Ginseng* / *GINO* offers query formulation in a controlled language akin to English. Therefore, the first three interfaces lie on the NL end of the Formality Continuum towards its middle. As such, they analyze a user query, match it to the content of a KB, and translate these matches into statements of a formal query language (i.e., SPARQL) in order to execute it. The last interface, *Semantic Crystal*, belongs to the formal approaches, as it exhibits a graphical query language. The guided and controlled entry overcomes the habitability problem of NL systems (providing a trade-off between structuredness and freedom) and ensuring all queries make sense in the context of the loaded KB. However, as stated in this usability study "users favor query languages that impose some structure but do not overly restrict them", thus, from the four systems, Querix was the interface preferred by the users, which query language (full English) was perceived as a natural, not formal, guiding structure.

The interface that has the least restrictive and most natural query language, **NLP-Reduce** [52], allows almost any NL input (from ungrammatical inputs, like keywords and sentence fragments, to full English sentences). It processes NL queries as bags of words, employing only two basic NLP techniques: stemming and synonym expansion. Essentially, it attempts to match the parsed question words to the synonym-enhanced triples stored in the lexicon (the lexicon is generated from a KB and expanded with WordNet synonyms), and generates SPARQL statements for those matches. It retrieves all those triples for which at least one of the question words occur as an object property or literal, favouring triples which cover most words and with best matches, and joins the resultant triples to cover the query.

The second interface **Querix** [53] is also a pattern matching NLI, however, the input is narrowed to full English (grammatically correct) questions, restricted only with regard to sentence beginnings (i.e., only questions starting with "which", "what", "how many", "how much", "give me" or "does"). In contrast with NLP-Reduce, Querix makes use of the syntactical structure of input questions to find better matches in the KB. Querix uses the Stanford parser to analyze the input query, then, from the parser's syntax tree, extended with WordNet synonyms, it identifies triple patterns for the query. These triple patterns are matched in the synonym-enhanced KB by applying pattern matching algorithms. When a KB is chosen, the RDF triples are loaded into a Jena model, using the Pellet reasoner to infer all implicitly defined triples and WordNet to produce synonym-enhanced triples. Pattern matching is done by searching for triples that include one of the nouns or verbs in the query. Querix does not try to resolve NL ambiguities, but asks the user for clarifications in a pop-up dialog menu window to disambiguate. Several triples can be retrieved for the nouns, verbs and their synonyms. Those that matches the query triples are selected, and from these, a SPARQL query is generated to be executed in the Jena's SPARQL engine.

In the middle of the formality continuum, **GINSENG** [8] controls a user's input via a fixed vocabulary and predefined sentence structures through menu-based options, as such it falls into the category of *guided input NL* interfaces, similar to **LingoLogic** [91]. These systems do not try to understand NL queries but they use menus to specify NL queries in small and specific domains. GINSENG uses a small static grammar that is dynamically extended with elements from the loaded ontologies and allows an easy adaptation to new ontologies, without using any predefined lexicon beyond the vocabulary that is defined in the static sentence grammar and provided by the loaded ontologies. When the user enters a sentence, an incremental parser relies on the grammar to constantly (1) propose possible continuations to the sentence, and (2) prevent entries that would not be grammatically interpretable.

PANTO [96] is a portable NLI that takes a NL question as input and executes a corresponding SPARQL query on a given ontology model. It relies on the statistical Stanford parser to create a parse tree of the query from which triples are generated. These triples are mapped to the triples in the lexicon. The lexicon is created when a KB is loaded into the system, by extracting all entities enhanced with WordNet synonyms. Following the AquaLog model, it uses two intermediate representations: the Query-Triples, which rely solely on the linguistic analysis of the query sentence, and the Onto-Triples that match the query triples and are extracted using the lexicon, string distance metrics and WordNet. PANTO can handle conjunctions / disjunctions, negation, comparatives and superlatives (those that can be interpreted with *Order by* and *Limit* on *datatype*, superlatives that require the functionality *count* are not supported).

Similarly, in **QuestIO** [88] NL queries are translated into formal queries but the system is reliant on the use of gazetteers initialized for the domain ontology. In QuestIO users can enter queries of any length and form. QuestIO works by recognizing concepts inside the query through the gazetteers, without relying on other words in the query. It analyzes potential relations between concept pairs and ranks them according to string similarity measures, the specifity of the property or distance between terms. QuestIO supports conjunction and disjunction.

FREyA [21] is the successor to QuestIO, providing improvements with respect to a deeper understanding of a question's semantic meaning, to better handle ambiguities when ontologies are spanning diverse domains. FREyA allows users to enter queries in any form. Therefore, to identify the answer type of the question and present a concise answer to the user a syntactic parse tree is generated using the Stanford parser. In addition, FREyA assists the user to formulate a query through the generation of clarification dialogs; the user's selections are saved and used for training the system in order to improve its performance over time for all users. Similar to AquaLog's learning mechanism, FREyA uses ontology reasoning to learn more generic rules, which could then be reused for the questions with similar context (e.g., for the superclasses of the involved classes).

Given a user query, the process starts with finding ontology-based annotations in the query, if there are ambiguous annotations that cannot be solved by reasoning over the context of the query (e.g., "Mississippi" can be a river or a state) the user is engaged in a dialog. The quality of the annotations depends on the ontology-based gazetteer OntoRoot, which is the component responsible for creating the annotations. The suggestions presented to the user in the clarification dialogs have an initial ranking based on synonym detection and string similarity. Each time a suggestion is selected by the user, the system learns to place the correct suggestions at the top for any similar question. These dialogs also allow translating any additional semantics into the relevant operations (such is the case with superlatives, which cannot be automatically understood without additional processing, i.e., applying a maximum or minimum function to a datatype property value). Triples are generated from the ontological mappings taking into account the domain and range of the properties. The last step is generating a SPARQL query by combining the set of triples.

We have selected a representative selection of state-of-the-art NL interfaces over ontologies to understand the advances and limitations in this area. However, this study is not exhaustive[9], and other similar systems to structured knowledge sources exist, such as **ONLI** [76], a QA system used as front-end to the RACER reasoner. ONLI transform the user NL queries into a nRQL query format that supports the <argument, predicate, argument> triple format. It accepts queries with quantifiers and number restrictions. However, from [76] it is not clear how much effort is needed to customize the system for different domains. [27] also developed a NL interface for a Web-based **tourism platform**. The system uses an ontology that describes the domain, the linguistic relationships between the domain concepts, and parameterised SQL fragments used to build the SQL statements representing the NL query. A lightweight grammar analyzes the question to combine the SQL statements accordingly. The system was online for ten days and collected 1425 queries (57.05% full input queries and the rest were keywords and question fragments). Interestingly, this study shows that the complexity of the NL questions collected was relatively low (syntactically simple queries combining an average of 3.41 concepts), and they can be parsed with shallow grammars.

Another approach with elaborated syntactic and semantic mechanisms that allows the user to input full NL to query KBs was developed by [34], **Frank et al. system** applies deep linguistic analysis to a question and transforms it into an ontology-independent internal representation based on conceptual and semantic characteristics. From the linguistic representation, they extract the so-called *proto queries*, which provide partial constraints for answer extraction from the underlying knowledge sources. Customization is achieved through hand-written rewriting rules transforming FrameNet like structures to domain-specific structures as provided by the domain ontology. A prototype was implemented for two application domains: the Nobel prize winners and the language technology domains, and was tested with a variety of question types (wh-, yes-no, imperative, definition, and quantificational questions), achieving precision rates of 74.1%.

To cope with the slower pace of increase in new knowledge in semantic repositories, in compassion with non-semantic Web repositories, **SemanticQA** [90] makes it possible to complete partial answers from a given ontology with Web documents. SemanticQA assists the users in constructing an input question as they type, by presenting valid suggestions in the universe of discourse of the selected ontology, whose content has been previously indexed with Lucene. The matching of the question to the ontology is performed by exhaustively matching all word combinations in the question to ontology entities. If a match is not found, WordNet is also used. Then all generated ontological triples are combined into a single SPARQL query. If the SPARQL query fails, indicating that some triples have no answers in the ontology, the system attempts to answer the query by searching in the snippets returned by Google. The collection of keywords passed to Google is gathered from the labels of the ontological entities plus WordNet. The answers are ranked using a semantic answer score, based on the expected type (extracted from the ontology) and the distance between all terms in the keyword set. To avoid ambiguity it allows restricting the document search to a single domain (e.g., PubMed if the user is looking for bio-chemical information). A small scale ad-hoc test was performed with only eight samples of simple factoid questions using the Lehigh University Benchmark ontology[10] (63% precision),

[9] See, for example, the EU funded project QALL-ME on multimodal QA: http://qallme.fbk.eu/

[10] http://swat.cse.lehigh.edu/projects/lubm/

and six sample queries using the SwetoDblp ontology (83% precision) [1].

One can conclude that the techniques used to solve the lexical gap between the users and the structured knowledge are largely comparable across all systems: off-the-shelf parsers and shallow parsing are used to create a triple-based representation of the user query, while string distance metrics, WordNet, and heuristics rules are used to match and rank the possible ontological representations.

4.2. Limitations of domain-specific QA approaches on the large SW

Most of the semantic QA systems reviewed in this paper are portable or agnostic to the domain of the ontology, even though, in practice they differ considerably in the degree of domain customization they require. Regardless of the various fine-grained differences between them, most ontology-aware systems suffer from the following main limitation when applied to a Web environment: ***they are restricted to a limited set of domains***. Such domain restriction may be identified by the use of just one, or a set of, ontology(ies) covering one specific domain at a time, or the use of one large ontology which covers a limited set of domains. The user still needs to tell these systems which ontology is going to be used. For instance, in AquaLog the user can select one of the pre-loaded ontologies or load a new ontology into the system (to be queried the ontology is temporarily stored in a Sesame store in memory). Like in NLIDB, the key limitation of all the aforementioned systems is the one already pointed out in [45], with the exception of FREyA (see Section 4.3) these systems presume that the knowledge the system needs to answer a question is limited to the knowledge encoded in one, or a set of homogeneous ontologies at a time. Therefore, they are essentially designed to support QA in corporate databases or *semantic intranets*, where a shared organizational ontology (or a set of them) is typically used to annotate resources. In such a scenario ontology-driven interfaces have been shown to effectively support the user in formulating complex queries, without resorting to formal query languages. However, these systems remain brittle, and any information that is either outside the semantic intranet, or simply not integrated with the corporate ontology remains out of bounds.

As a result, it is difficult to predict the feasibility of these models to scale up to open and heterogeneous environments, where an unlimited set of topics is covered. Nonetheless, we detail next the intrinsic characteristics of these systems, which in principle impair their suitability to scale up to the open SW in the large:

Domain-specific grammar-based systems: In these systems grammars are used to syntactically analyze the structure of a NL query and interpret, if there are no linguistic ambiguities, how the terms in a query link to each other. According to [18] it is difficult to devise grammars that are sufficiently expressive. Often, they are quite limited with regard to the syntactic structures they are able to understand or are domain dependent (although grammars can also be fully domain independent, as it is the case with AquaLog). Nevertheless, according to [60] users tend to use a limited language when interacting with a system interface, so grammars do not need to be complete. Systems like ORAKEL that involve the user in the difficult task of providing a domain-specific grammar are not a suitable solution in a multi-ontology open scenario.

Pattern-matching or bag-of-words approaches: These systems search for the presence of constituents of a given pattern in the user query. As stated in [51] "the more flexible and less controlled a query language is, the more complex a system's question analyzing component needs to be to compensate for the freedom of the query language". However, naïve and flexible pattern-matching systems work well in closed scenarios, like the NLP-Reduce system, in which complexity is reduced to a minimum by only employing two basic NLP techniques: stemming and synonym expansion. Their best feature is that they are ontology independent and even ungrammatical and ill-formed questions can be processed. Nevertheless, their little semantics and lack of sense disambiguation mechanisms hamper their scalability to a large open scenario. In a non-trivial scenario, pattern-matching or bag-of-words approaches (QACID, QuestIO), together with the almost unlimited freedom of the NL query language, result in too many possible interpretations of how the words relate together. Thus, increasing the risk of not finding correct (SPARQL) translations and suffering from the habitability problem [50]. As stated in an analysis of semantic search systems in [44]: "Naïve approaches to semantic search are computationally too expensive and increase the number of results dramatically, systems thus need to find a way to reduce the search space".

Guided interfaces: Guided and controlled interfaces, like GINO, which generates a dynamic grammar rule for every class, property and instance and presents pop-up boxes to the user to offer all the possible completions to the user's query, are not feasible solutions in a large multi-ontology scenario. As stated in [50] when describing GINO "It is important to note that the vocabulary grows with every additional loaded KB, though users have signaled that they prefer to load only one KB at a time".

Disambiguation by dialogs and user interaction: Dialogs are a popular and convenient feature [51] to resolve ambiguous queries, for the cases in which the context and semantics of the ontology is not enough to choose an interpretation. However, to ask the user for assistance every time an ambiguity arises (AquaLog, Querix) can make the system not usable in a multi-domain scenario where many ontologies participate in the QA processes. In FREyA, the suggestions presented on the dialogs are ranked using a combination of string similarity and synonym detection with WordNet and Cyc[11]. However, as stated in [21]: "the task of creating and ranking the suggestions before showing them to the user is quite complex, and this complexity arises [sic] as the queried knowledge source grows".

Domain dependent lexicons and dictionaries: High performance can be obtained with the use of domain dependent dictionaries at the expense of portability (as in the e-librarian system). However it is not feasible to manually build, or rely on the existence of domain dictionaries in an environment with a potentially unlimited number of domains.

Lexicons generated on demand when a KB is loaded: The efficiency of automatically generating triple pattern lexicons when loading an ontology (PANTO, NLP-Reduce, QuestIO, FREyA), including inferred triples formed applying inference rules and WordNet lexically related words independently of their sense, decreases with the size of the ontology and is itself a challenging issue if multiple large-scale ontologies are to be queries simultaneously. In contrast with the structured indexes used by PANTO or NLP-Reduce, entity indexes can benefit from less challenging constraints in terms of index space, creation time and maintenance. However, ignoring the remaining context provided by the query terms can ultimately lead to an increase in query execution time to find the adequate mappings.

4.3. Open QA over the Semantic Web

Latest research on QA over the SW focuses on overcoming the domain-specific limitations of previous approaches. The importance of the challenge, for the SW and also NLP communities, to scale QA approaches to the open Web, i.e., Linked Data, has been recognized by the appearance of the first evaluation challenge for QA over Linked Data in the 1st workshop on QA over Linked Data (QALD-1)[12].

From the QA systems analyzed in 4.1, FREyA is currently the only one able to query large, heterogeneous and noisy single sources (or ontological graph) covering a variety of domains, such as DBpedia [10].

Similarly, moving into the direction of suitable systems for open domain QA systems, PowerAqua [66] evolved from the AquaLog system presented in Section 4.1, which works using a single ontology, to the case of multiple heterogeneous ontologies. PowerAqua is the first system to perform QA over structured data in an open domain scenario, allowing the system to benefit, on the one hand from the combined knowledge from the wide range of ontologies autonomously created on the SW, reducing the knowledge acquisition bottleneck problem typical of KB systems, and on the other hand, to answer queries that can only be solved by composing information from multiple sources.

PowerAqua follows a pipeline architecture, the query is first transformed by the linguistic component into a triple based intermediate format, or Query-Triples, in the form <subject, property, object>. At the next step, the Query-Triples are passed on to the PowerMap mapping component [65], which identifies potentially suitable semantic entities in various ontologies that are likely to describe query terms and answer a query. PowerMap uses both WordNet and the SW itself (owl:sameAs) to find synonyms, hypernyms, derived words, meronyms and hyponyms. In the third step, the Triple Similarity Service, exploring the ontological relations between these entities, matches the Query-Triples to ontological expressions specific to each of the considered semantic sources, producing a set of Onto-Triples that jointly cover the user query, from which answers are derived as a list of entities matching the given triple patterns in each semantic source. Finally, because each resultant Onto-Triple may only lead to partial answers, they need to be combined into a complete answer. The fourth component merges and ranks the various interpretations produced in different ontologies. Among other things, merging requires the system to identify entities denoting the same individual across ontologies. Once answers are merged, ranking, based on the

[11] http://sw.opencyc.org

[12] http://www.sc.cit-ec.uni-bielefeld.de/qald-1

quality of mappings and popularity of the answers, can also be applied to sort the answers. As shown in [63], merging and ranking algorithms enhance the quality of the results with respect to a scenario in which merging and ranking is not applied.

To scale PowerAqua model to an open Web environment, exploiting the increasingly available semantic metadata in order to provide a good coverage of topics, PowerAqua is coupled with: a) the Watson SW gateway, which collects and provides fast access to the increasing amount of online available semantic data, and b) its own internal mechanism to index and query selected online ontological stores, as an alternative way to manage large repositories, like those offered by the Linked Data community, often not available in Watson due to their size and format (RDF dumps available as compressed files).

4.4. Performance of ontology-based QA systems based on their state-of-the-art evaluations

We examine the performance of the ontology-based QA systems previously presented by looking at the evaluation results carried out in the literature. In contrast to the IR community, where evaluation using standardized techniques, such as those used for TREC competitions, has been common for decades, systematic and standard evaluation benchmarks to support independent datasets and comparisons between systems are not yet in place for semantic QA tools. Important efforts have been done recently towards the establishment of common datasets, methodologies and metrics to evaluate semantic technologies, e.g., the SEALS project[13] to assess and compare different interfaces within a user-based study in a controlled scenario. However, the diversity of semantic technologies and the lack of uniformity in the construction and exploitation of the data sources are some of the main reasons why there is still not a general adoption of evaluation methods. Therefore evaluations are generally small scale with ad-hoc tasks that represent the user needs and the system functionality to be evaluated [69,95]. Although the different evaluation set-ups and techniques undermine the value of direct comparisons, nevertheless, they are still useful to do an approximate assessment of the strength and weaknesses of the different systems. We hereby briefly describe the different evaluation methods and performance results. These are presented in Table 2.

Evaluations performed in the early days of the SW had to cope with the sparseness and limited access to high quality and representative public semantic data. As a result, to test the AquaLog system [67] two (manually built) rich ontologies were used and the query sets were gathered from 10 users. This approach gave a good insight about the effectiveness of the system and the extent to which AquaLog satisfied user expectations about the range of queries it is able to answer across two different domains. In order for an answer to be correct, AquaLog had to correctly align the vocabularies of both the asking query and the answering ontology. The test showed a 63.5% success, a promising result considering that almost no linguistic restrictions were imposed on the questions. Because of the sequential nature of the AquaLog architecture, failures were classified according to which component caused the system to fail. The major limitations were due to lack of appropriate reasoning services defined over the ontology (e.g., temporal reasoning, quantifier scoping, negations – “not”, “other than”, “except”), comparatives and superlatives, a limited linguistic coverage (e.g., queries that were too long and needed to be translated into more than two triples), and lack of semantic mechanisms to interpret a query given the constraints imposed by the ontology structures (e.g., AquaLog could not properly handle anaphoras[14], compound nouns, non-atomic semantic relations, or reasoning with literals).

Alternatively, the evaluations presented in [50] for NLP Reduce, Querix and Ginseng were measured with the standard IR performance metrics: precision and recall. Failures are categorized according to whether they are due to: 1) “no semantically tractable queries” [85,89], i.e., questions that were not accepted by the query languages of the interfaces or 2) irrelevant SPARQL translations. Recall was defined as the number of questions from the total set that were correctly answered (% success), while precision is the number of queries that were correctly matched to a SPARQL query with respect to the number of semantically tractable questions (see Fig. 2). Thus, the average recall values are lower than the precision values, a logical consequence of the fact that recall is

[13] Campaign 2010 results at: http://www.seals-project.eu/seals-evaluation-campaigns/semantic-search-tools/results-2010

[14] A linguistic phenomenon in which pronouns (e.g. “she”, “they”), and possessive determiners (e.g. “his”, “theirs”) are used to implicitly denote entities mentioned in an extended discourse (freepatentsonline.com/6999963.html).

Table 2

Performance results of the ontology-based QA systems evaluated in the state of the art

	Datasets	Nº queries	% Success (S)		Domain independent
AquaLog	KMi semantic portal[15]	69	**58**%(S)	63.5%	Yes (NL queries)
	Wine and food[16]	68	**69.11**%(S)		
NLP Reduce	Geography	887	95.34%(P)/ **55.98**%(S)	55.3%	Yes (NL and keyword queries)
	Restaurants	251	80.08%(P)/ **97.10**%(S)		
	Jobs	620	81.14%(P)/ **29.84**%(S)		
Querix	Geography (USA)	887	91.38%(P)/ **72.52**%(S)	54.4%	Yes (NL wh-queries)
	Restaurants	251	94.31%(P)/ **59.36**%(S)		
	Jobs	620	80.25(P)/ **31.45**%(S)		
Ginseng	Geography (USA)	887	98.86%(P)/ **39.57%**(S)	48.6%	Yes (guided interface)
	Restaurants	251	100%(P)/ **78.09**%(S)		
	Jobs	620	97.77%(P)/ **28.23**%(S)		
PANTO	Geography (USA)	877 out 880	88.05%(P)/ 85.86%(R)= **75.6**%(S)	80%	Yes (NL queries)
	Restaurants	238 out of 250	90.87%(P)/ 96.64%(R)= **87.8**%(S)		
	Jobs	517 out of 641	86.12%(P)/ 89.17%(R)= **76.8**%(S)		
ORAKEL	Geography (Germany)	454	**93**%		Domain-dependent grammar (NL queries)
QuestIO	GATE ontology	22	**71.88**%		Yes (NL queries)
	Travel guides	Not reported			
e-Librarian	Computer history and mathematics	229	**97**%		Domain-dependent dictionary (NL queries)
QACID	Cinema	100	**80**%		Domain-dependent collection NL queries
FREyA	Geography	250	**92.4**%		Yes

based on the number of semantically tractable questions (those that the system can transform into SPARQL queries, independently of whether the query produced is appropriate or not). For instance Ginseng has the highest precision but the lowest recall and semantic tractability due to its limited query language (some of the full NL test queries could not be entered into the system). Also, the use of comparative and superlative adjectives in many of the questions decreased the semantic tractability rate in NLP–Reduce, which cannot process them. To enable a comparison, these NLIs were benchmarked with the same three externally sourced test sets with which other NLI systems (PANTO by Wang et al. and the NLIDBs PRECISE by Popescu et al.) had already been evaluated. These three datasets are based on the *Mooney NL Learning Data* provided by Ray Mooney and his group from the University of Texas at Austin [89] and translated to OWL for the purposes of the evaluation in [50]. Each dataset supplies a KB and set

[15] The akt ontology: http://kmi.open.ac.uk/projects/akt/ref-onto/

[16] W3C, OWL Web Ontology Language Guide: http://www.w3.org/TR/2003/CR-owl-guide-0030818/

$$\text{recall} = \frac{\text{number of correct SPARQL queries produced}}{\text{total number of questions}}$$

$$\text{precision} = \frac{\text{number of correct SPARQL queries produced}}{\text{number of semantically tractable questions}}$$

Fig. 2. Definition of precision and recall by [50].

of English questions, belonging to one of the following domains: geography (9 classes, 28 properties and 697 instances), jobs (8 classes, 20 properties, 4141 instance) and restaurants (4 classes, 13 properties and 9749 instances).

PANTO assesses the rate of how many of the translated queries correctly represent the semantics of the original NL queries by comparing its output with the manually generated SPARQL queries. The metrics used are precision and recall, defined in [96] as "precision means the percentage of correctly translated queries in the queries where PANTO produced an output; recall refers to the percentage of queries where PANTO produced an output in the total testing query set". Note that these definitions make the notion of correctness somewhat subjective, even between apparently similar evaluations. Recall is defined differently in PANTO and the approaches in [50]. For [50] recall is the number of questions from the total correctly answered, which is defined as a *%success* in AquaLog, while for PANTO is the number of questions from the total that produce an output, independently of whether the output is valid or not. Thus, to measure %success (how many NL question the system successfully transformed in SPARQL queries) in PANTO we need to multiply precision by recall and divide it by 100; the results are in Tabel 2. There are also some discrepancies in the number of queries in the Mooney datasets between [50] and [96].

QuestIO was tested on a locally produced ontology, generated from annotated postings in the GATE mailing list, with 22 real user queries that could be answered in the ontology and a Travel Guides Ontology with an unreported number of queries, to demonstrate portability. The initialization time of QuestIO with the Travel Guides ontology (containing 3194 resources in total) was reported to be 10 times longer, which raises some concerns in terms of scalability. A query is considered correctly answered if the appropriate SeRQL query is generated (71.8% success).

FREyA was also evaluated using 250 questions from the Mooney geography dataset. Correctness is evaluated in terms of precision and recall, defined in the same way as in [50]. The ranking and learning mechanism was also evaluated, they report an improvement of 6% in the initial ranking based on 103 questions from the Mooney dataset. Recall and precision values are very high, both reaching 92.4%.

The system that reports the highest performance is the e-Librarian: in an evaluation with 229 user queries 97% were correctly answered, and in nearly half of the questions only one answer, the best one, was retrieved. Two prototypes were used: a computer history expert system and a mathematics expert system. The higher precision performance of e-Librarian with respect to a system like PANTO reflects the difficulty with precision performance on completely portable systems.

QACID has been tested with an OWL ontology in the cinema domain, where 50 users were asked to generate 500 queries in total for the given ontologies. From these queries, 348 queries were automatically annotated by an Entity Annotator and queries with the same ontological concepts were grouped together, generating 54 clusters that were manually associated to SPARQL queries. The results reported in an on-field evaluation, where 10 users were asked to formulate spontaneous queries about the cinema domain (a total of 100 queries), show an 80% of precision.

As already mentioned, the different evaluation setups and techniques undermine the validity of direct comparisons, even for similar evaluations, like the ones between PANTO and the systems in [50], because of the different sizes of the selected query samples and the different notions of evaluating correctness.

These performance evaluations share in common the pattern of being ad-hoc, user-driven and using unambiguous, relatively small and good quality semantic data. Although they test the feasibility of developing portable NLIs with high retrieval performance, these evaluations also highlight that the NLIs with better performance usually tend to require a degree of expensive customization or training. As already pointed out in [22], to bridge the gap between the two extremes, domain independency and performance, the quality of the semantic data have to be very high, to ensure a good lexicalization of the ontology and KBs and a good coverage of the vocabulary. Nonetheless, as previously reported in AquaLog, and recently evaluated in FREyA, the inclusion of a learning mechanism offers a good trade-off between user interaction and performance, ensuring an increase in performance over time by closing the lexical gap between users and ontologies, without compromising portability.

Large ontologies pose additional challenges with respect to usability, as well as performance. The ontologies used in the previous evaluations are relatively small; allowing to carry out all processing operations in memory, thus, scalability is not evaluated.

Linked Data initiatives are producing a critical mass of semantic data, adding a new layer of complexity in the SW scenario, from the exploitation of small domain specific ontologies to large generic open domain data sources containing noisy and incomplete data. Thus, two main user-centric evaluations have been conducted to test PowerAqua: before and after using Linked Data, to investigate whether it can be used to exploit the data offered by Linked Data. In the first evaluation [66], PowerAqua was evaluated with a total of 69 queries, generated by 7 users, that were covered by at least one ontology in the semantic information space (consisting in more than 130 Sesame repositories, containing more than 700 ontological documents). PowerAqua successfully answered 48 of these questions (69.5%). The second evaluation was focused on scalability and performance when introducing into the previous evaluation setup one of the largest and most heterogeneous datasets in Linked Data, DBpedia [64]. The time needed to answer a query depends on two main factors: (1) the total number of (SPARQL-like) calls send to the ontologies to explore relevant connections between the mappings, which depends directly on the number of semantic sources and mappings that take part in the answering process, and (2) the response times to these calls, which depends on the complexity of the (SPARQL) queries and the size of the ontology. PowerAqua algorithms were optimized by introducing heuristics to balance precision and recall, thus to analyze the most likely solutions first (iteratively refining candidates only as needed). These heuristics reduced by 40% in average the number of queries sent to the ontologies, however the response times to answer a query increased from 32 to 48 secs. Initial experiments using a different back-end for large-scale sources, i.e. Virtuoso instead of Sesame, reduced the average time to 20 secs. PowerAqua usability as a NL interface to semantic repositories, has also been evaluated following the formal benchmark proposed in SEALS 2010 [62], focused on the usability aspects of different search tools (in particular keyword-based, form-based and NL) within a controlled user study using the Mooney geography dataset. Of the systems tested, PowerAqua was the system with better usability results, evaluated as "good" by the users.

4.5. The competences of ontology-based QA systems

The main clear advantage of the use of NL query tools is the easy interaction for non-expert users. As the SW is gaining momentum, it provides the basis for QA applications to exploit and reuse the structured knowledge available on the SW. Beyond the commonalities between all forms of QA (in particular for the question analysis), in this section, we analyze the competencies of ontology-based QA with respect to the main traditional forms of QA.

4.5.1. Ontology-based QA with respect to NLIDB

Since the development of the first QA systems [3], there have been major improvements in the availability of lexical resources, such as WordNet; string distance metrics for name-matching tasks [17]; shallow, modular and robust NLP systems, such s GATE [20]; and NLP Parsers, such as the Stanford parser. In comparison with the latest work on NLIDB, the benefits of ontology-based QA are:

- **Ontology independence:** Later NLIDB systems [18] use intermediate representations to have a portable front end with general purpose grammars, while the back end is dependent on a particular database. As a result, long configuration times are normally required to port the system to a new domain. Ontology-based QA systems have successfully solved the portability problem, as the knowledge encoded in the ontology, together with (often shallow) domain-independent syntactic parsing, are the primary sources for understanding the user query, without the need to encode specific domain-dependent rules. Hence, these systems are practically ontology independent, less costly to produce, and require little effort to bring in new sources (AquaLog, PANTO, Querix, QuestIO, FREyA). Optionally, on these systems manual configuration or automatic learning mechanisms based on user feedback can optimize performance.
- **Able to handle unknown vocabulary in the user query:** NLIDB systems, such as PRECISE [85], require all the tokens in a query to be distinct and questions with unknown words are not semantically tractable. In ontology-based QA if a query term is lexically dissimilar from the vocabulary used by the ontology, and it does not appear in any manually or automatically created lexicon, studying the ontology "neighborhood" of the other terms in the query may lead to the value of the term or relation we are looking for.

In many cases this would be all the information needed to interpret a query.

- **Deal with ambiguities:** When ontologies are directly used to give meaning to the queries expressed by the user and retrieve answers, the main advantage is the possibility to link words to obtain their meaning based on the ontological taxonomy and inherit relationships, and thus, to deal with ambiguities more efficiently.

Summing up, the main benefits of ontology-based QA systems are that they make use of the semantic information to interpret and provide precise answers to questions posed in NL and are able to cope with ambiguities in a way that makes the system highly portable.

4.5.2. Ontology-based QA with respect to QA on text

Although most of the state-of-the-art of ontology-based QA still presumes that the knowledge needed is encoded in one ontology in a closed domain scenario, we envision ontology-based QA to move towards an open SW scenario, to become complementary to free-text open QA. While the first targets the open, structured SW to give precise answers, the second targets unstructured documents on the Web. Under such a perspective, a document search space is replaced by a semantic search space composed of a set of ontologies and KBs, providing a new context in which the results from traditional open QA can be applied. Although linguistic and ambiguity problems are common in most kinds of NL understanding systems, building a QA system over the SW has the following advantages:

- **Balancing relatively easy design and accuracy:** As seen in Section 3.2 the current state of the art open systems to query documents on the Web require sophisticated syntactic, semantic and contextual processing to construct an answer, including NE recognition [42]. These open QA systems classify queries using hierarchies of question types based on the types of answers sought (e.g., person, location, date, etc.) and filter small text fragments that contain strings with the same type as the expected answers [78,87]. In ontology-based QA there is no need to build complex hierarchies, to manually map specific answer types to WordNet conceptual hierarchies or to build heuristics to recognize named entities, as the semantic information needed to determine the type of an answer is in the publicly available ontology(ies). As argued in [80] a major difference between open-domain QA and ontology-based QA is the existence of domain-dependent information that can be used to improve the accuracy of the system.
- **Exploiting relationships for query translation:** NE recognition and IE are powerful tools for free-text QA (Section 3.2.1), although these methods scale well discovering relationships between entities is a crucial problem [87]. IE methods do not often capture enough semantics, answers hidden in a form not recognized but the patterns expected by the system could be easily disregarded, and one cannot always rely on WordNet coverage to determine the answer type or the type of the object of the verb in the question [83]. On the contrary, QA systems over semantic data can benefit from exploiting the explicit ontological relationships and the semantics of the ontology schema (e.g., type, subclassOf, domain and range), to understand and disambiguate a query. WordNet is only used for query expansion, to bridge the gap between the vocabulary of the user and the ontology terminology through lexically related words (such as synonyms).
- **Handling queries in which the answer type is unknown:** *What* queries, in which the type of the expected answer is unknown, are harder than other types of queries when querying free text [47]. However, the ontology simplifies handling what-is queries because the possible answer types are constrained by the types of the possible relations in the ontology.
- **Structured answers are constructed from ontological facts:** Arbitrary query concepts are mapped to existing ontology entities, answers are then obtained by extracting the list of semantic entities that comply with the facts, or fulfill the ontological triples or SPARQL queries. The approach to answer extraction in text-based QA requires first identifying entities matching the expected answer in text, e.g., using the WordNet mapping approach. Second, the answers within these relevant passages are selected using a set of proximity-based heuristics, whose weights are set by a machine-learning algorithm [83]. Although IR methods scale well, valid answers in documents that do not follow the syntactic patterns expected by the QA system can be easily disregarded.
- **Combining multiple pieces of information:** Ontological semantic systems can exploit the

power of ontologies as a model of knowledge to give precise, focused answers, where multiple pieces of information (that may come from different sources) can be inferred and combined together. In contrast, QA systems over free text cannot do so, as they retrieve pre-written paragraphs of text or answer strings (typically NPs or named entities) extracted verbatim from relevant text [83].

4.5.3. Ontology-based QA with respect to proprietary QA

It is costly to produce the large amounts of domain background knowledge, which are required by the proprietary open domain approaches described in Section 3.3. Although based on semantics, these systems do not reuse or take fully advantage of the freely available structured information on the SW. This is a key difference as they impose an internal structure on their knowledge and claim ownership of a trusted and curated homogeneous KB, rather than supporting the user in exploring the increasing number of distributed knowledge sources available on the Web.

4.6. Open research issues on open QA on the SW

Evaluations in [64] considered the results encouraging and promising, if one considers the openness of the scenario, and probe, to some extend, the feasibility and potential of the approach. Nonetheless, several issues remain open to any approach that wishes to benefit from exploiting the vast amount of emerging open Web data to elicit the most accurate answer to a user query:

- Heterogeneity and openness: the high ambiguity in the sources means that it is not always possible to have enough context to focus on precision when, because of heterogeneity, there are many alternative translations and interpretations to a query. For example, the main issue for PowerAqua is to keep real time performance in a scenario of perpetual change and growth, in particular when both very large heterogeneous sources from the Linked Data cloud, or thousands of small RDF sources from crawled data from Watson are added [62].
- Dealing with scalability as well as knowledge incompleteness: filtering and ranking techniques are required to scale to large amounts of data. There are often a huge number (from hundreds to thousands in many cases) of potential ontological hits with different meanings (domains), across and within the same dataset, that can syntactically map the terms in a user query. It is unfeasible to explore all possible solutions to obtain semantically sound mappings, however, filtering and domain-coverage heuristics to shift focus onto precision require making certain assumptions about quality of sources. If filtering heuristics are too strict, recall is affected in a noisy environment, where sources contain redundant and duplicated terms and incomplete information, either because not all ontological elements are populated at the level of instances or because of a lack of schema information (no domain and range for properties, or type for classes, difficult to parse literals, etc.).
- Sparseness: the potential is overshadowed by the sparseness and incompleteness of the SW when compared to the Web [84]. During the search process, it may happen that a) there are no available ontologies that cover the query, or b) there are ontologies that cover the domain of the query but only contain parts of the answer.

5. Related work on open user-friendly querying interfaces for the SW

In the previous sections, we have seen that QA systems have proven to be ontology independent or easily adaptable to new domains, while keeping their efficiency and retrieval performance even when shallow NLP techniques are used. By opening up to the SW scenario, these systems can reach their full potential and enhance or complement traditional forms of QA. In this section we broaden our scope and look at user-friendly semantic search systems and Linked Data querying interfaces, in search for models, beyond NL QA systems, that can in principle scale enough to open up, and even integrate, heterogeneous data sources on the Web of Data.

Many approaches exist to translate user queries into formal queries. Semantic search, a broader area than semantic QA, faces similar challenges to those tackled by QA systems when dealing with heterogeneous data sources on the SW. Here, we look at the solutions proposed in the literature for semantic search and how they address semantic heterogeneity from early information systems to the latest approaches to searching the SW. We further discuss how all QA approaches presented till now and the SW user-friendly querying models presented in this section are compared according to the criteria presented in Section 2, and how both research directions can converge into large scale open ontology-based QA for the SW, to solve the bottlenecks and limitations of both.

5.1. Early global-view information systems

The idea of presenting a conceptually unified view of the information space to the user, the “world-view”, has been studied in [58]. In early global information systems with well-defined boundaries, the solutions for interfacing and integrating heterogeneous knowledge sources, in order to answer queries that the original sources alone were unable to handle, are based on two approaches [80]: either all the information from multiple sources is extracted to create a unified database, or the set of databases can be seen as a federated database system with a common API, as in [5]. However, this type of centralized solution that forces users and systems to subscribe to a single ontology or shared model are not transferable to the open-world scenario, where the distributed sources are constantly growing and changing. The manual effort needed to maintain any kind of centralized, global shared approach for semantic mapping is not only very costly, in terms of maintaining the mapping rules in a highly dynamic environment [72], but it also has the added difficulty of “negotiating” a shared model, or API, that suits the needs of all the parties involved [11].

Lessons and remaining open issues: Interestingly, the problems faced by these early information systems are still present nowadays. Linked Data assumes re-use of identifiers and the explicit specification of strong inter-dataset linkage in an open distributed fashion, without forcing users to commit to an ontology. However, on the SW the heterogeneity problem can hardly be addressed only by the specification of mapping rules. As stated in [84], “although RDF theoretically offers excellent prospects for automatic data integration assuming re-use of identifiers and strong inter-dataset linkage, such an assumption currently only weakly holds”. Therefore, open semantic applications need to handle heterogeneity and mappings on the fly, in the context of a specific task.

5.2. Evolution of semantic search on the Web of data

Aiming to overcome the limitations of keyword-based search, semantic search has been present in the IR field since the eighties [19], through the use of domain knowledge and linguistic approaches (thesaurus and taxonomies) to expand user queries. Ontologies were soon envisaged as key elements to represent and share knowledge [36] and enable a move beyond the capabilities of current search technologies [37]. As stated by [30] “the most common way in which semantic search has been addressed is through the development of search engines that execute a user query in the KB, and return tuples of ontology values which satisfy the user request”.

A wide-ranging example is TAP [39], one of the first keyword-based semantic search systems, which presented a view of the search space where documents and concepts are seen as nodes in a semantic network. In TAP the first step is to map the search term to one or more nodes of the SW. A term is searched by using its rdfs:label, or one of the other properties indexed by the search interface. In ambiguous cases it chooses a search term based on the popularity of the term (frequency of occurrence in a text corpus), the user profile, the search context, or by letting the user pick the right denotation. The nodes that express the selected denotation of the search term provide a starting point to collect and cluster all triples in their vicinity (the intuition being that proximity in the graph reflects mutual relevance between nodes).

In 2004 the annual SW Challenge was launched, whose first winner was CS Aktive Space [86]. This application gathers and combines a wide range of heterogeneous and distributed Computer Science resources to build an interactive portal. The top two ranked entries of the 2005 challenge, Flink [73] and Museum Finland [48], are similar to CS Aktive Space as they combine heterogeneous and distributed resources to derive and visualize social networks and to expose cultural information gathered from several museums respectively. However, there is no semantic heterogeneity and “openness” in them: these tools simply extract information, scraped from various relevant sites, to populate a single, pre-defined ontology. A partial exception is Flink, which makes use of some existing semantic data, by aggregating online FOAF files.

Later semantic systems adopted interesting approaches to query interpretation, where keyword queries are mapped and translated into a ranked list of formal queries. These include SemSearch [57], XXPloreKnow! [92] and QUICK [100]. For instance, SemSearch supports the search for semantic relations between two terms in a given semantic source, e.g., the query ‘news:PhD students’ results in all instances of the class news that are related to PhD students. SemSearch and XXPloreKnow! construct several formal queries for each semantic relation or combination of keywords’ matches, where ranking is used to identify the most relevant meanings of keywords, and to limit the number of different combinations. To go beyond the expressivity of keywords and translate a keyword query into a set of semantic queries that are most likely to ones intended by the user, QUICK computes all possible semantic queries among the keywords for the user to select one. With each selection the space of semantic interpretations is reduced, and the query is incrementally constructed by the user.

The approach in [29] combines standard Web search queries with ontological search queries. It assumes that Web pages are enriched with annotations that have unique identifiers and are relative to an underlying ontology. Web queries are then interpreted based on the underlying ontology, allowing the formulation of precise complex ontological conjunctive queries as SW search queries. Then these complex ontology queries are translated into sequences of standard Web queries answered by standard Web search. Basically, they introduce an offline ontological inference step to compute the completion of all semantic annotations, augmented with axioms deduced from the annotations and the background ontologies, as well as an online step that converts the formal conjunctive ontological queries into semantic restrictions before sending them to the search engine.

Different to previous approaches, restricted by a domain ontology, the system presented in [31] exploits the combination of information spaces provided by the SW and by the (non-semantic) Web, supporting: (i) semantic QA over ontologies and (ii) semantic search over non-semantic documents. First, answers to a NL query are retrieved using the PowerAqua system [66]. Second, based on the list of ontological entities obtained as a response to the user's query and used for query expansion, the semantic search over documents is accomplished by extending the system presented in [13] for annotating documents. The output of the system consists of a set of ontology elements that answer the user's question and a complementary ranked list of relevant documents. The system was evaluated reusing the queries and judgments from the TREC-9 and TREC 2001. However, at that time, only 20% of queries were partially covered by ontologies in the SW. For those queries, where semantic information was available, it led to important improvements over the keyword-based baseline approach, degrading gracefully when no ontology satisfied the query.

Lessons and remaining open issues: As argued in [81], the major challenge faced by early semantic applications was the lack of online semantic information. Therefore, in order to demonstrate their methods, they had to produce their own semantic metadata. As a result, the focus of these tools is on a single, well-defined domain, and they do not scale to open environments. The latest semantic applications, set out to integrate distributed and heterogeneous resources, even though these resources end up centralized in a semantic repository aligned under a single ontology. Therefore, these approaches follow the paradigm of smart KB-centered applications, rather than truly exploring the dynamic heterogeneous nature of the SW [81]. Furthermore, as discussed in [29], pressing research issues on approaches to semantic search on the Web are on the one hand, the ability to translate NL queries into formal ontological queries (the topic of this survey), and on the other hand, how to automatically add semantic annotations to Web content, or alternatively, extract knowledge from Web content without any domain restriction [31].

5.3. Large scale semantic search and Linked Data interfaces

New technologies have been developed to manipulate large sets of semantic metadata available online. Search engines for the SW collect and index large amounts of semantic data to provide an efficient keyword-based access point and gateway for other applications to access and exploit the growing SW. Falcons [14] allows concept (classes and properties) and object (instance) search. The system recommends ontologies on the basis of a combination of the TF-IDF technique and popularity for concept search, or the type of objects the user is likely to be interested in for object search. Falcons indexes 7 million of well-formed RDF documents and 4,400 ontologies [14]. Swoogle [26] indexes over 10,000 ontologies, Swoogle claims to adopt a Web view on the SW by using a modified version of the PageRank popularity algorithm, and by and large ignoring the semantic particularities of the data that it indexes. Later search engines such as Sindice [82] index large amounts of semantic data, over 10 billion pieces of RDF, but it only provides a *look-up* service that allows applications and users to locate semantic documents. Watson [23] collects the available semantic content from the Web, indexing over 8,300 ontologies, and also offers an API to query and discover semantic associations in ontologies at run time, e.g., searching for relationships in specific ontological entities. Indeed out of these four ontology search engines, only Watson allows the user to exploit the reasoning capabilities of the semantic data, without the need to process these documents locally. The other engines support keyword search but fail to exploit the semantic nature of the content they store and therefore, are still rather limited in their ability to support systems which aim to exploit online ontologies in a dynamic way [24].

Other notable exceptions to this limited-domain approach include search applications demonstrated in the Semantic Web Challenge competitions, and more recently the Billion Triples Challenge (btc)[17], aimed at stimulating the creation of novel demonstrators that have the capability to scale and deal with heterogeneous data crawled from the Web. Examples include SearchWebDB [97], the second prize-winner of the btc in 2008, which offers a keyword-based interface to integrated data sources available in the btc datasets. However, as keywords express the user needs imprecisely, the user needs to be asked to select among all possible interpretations. In this system the mappings between any pairs of data sources at the schema or data levels are computed a priori and stored in several indexes: the *keyword index*, the *structure index* and the *mapping index*. The disadvantage being that, in a highly dynamic environment, static mappings and complex structural indexes are difficult to maintain, and the data quickly becomes outdated.

The eRDF infrastructure [38] explores the Web of Data by querying distributed datasets in live SPARQL endpoints. The potential of the infrastructure was shown through a prototype Web application. Given a keyword, it retrieves the first result in Sindice to launch a set of SPARQL queries in all SPARQL end points, by applying an evolutionary anytime query algorithm, based on substitutions of possible candidate variables for these SPARQL queries. As such, it retrieves all entities related to the original entity (because they have the same type or a shared relationships to the same entity, for example Wendy Hall and Tim Berners Lee both hold a professorship at the university of Southampton).

Faceted views have been widely adopted for many RDF datasets, including large Linked Data datasets such as DBPedia, by using the Neofonie[18] search technology. Faceted views, over domain-dependent data or homogenous sources, improve usability and expressivity over lookups and keyword searches, although, the user can only navigate through the relations explicitly represented in the dataset. Faceted views are also available over large-scale Linked Data in Virtuoso [28], however scalability is a major concern, given that faceted interfaces become difficult to use as the number of possible choices grows. The ranking of predicates to identify important facets is obtained from text and entity frequency, while semantics associated with the links is not explored.

Mash-ups [93] are able to aggregate data coming from heterogeneous repositories and semantic search engines, such as Sindice, however these systems do not differentiate among different interpretations of the query terms, and disambiguation has to be done manually by the user.

Lessons and remaining open issues: these systems have the capability to deal with the heterogeneous data crawled from the Web. However, they have limited reasoning capabilities: mappings are either found and stored a priori (SearchWebdB), or disambiguation between different interpretations is not performed (eRDF). The scale and diversity of the data put forward many challenges, imposing a trade-off between the complexity of the querying and reasoning process and the amount of data that can be used. Expressivity is also limited compared to the one obtained by using query languages, which hinders the widespread exploitation of the data Web for non-expert users. Finally, in both facets and mash-ups, the burden to formulate queries is shifted from the system to the user. Furthermore, they do not perform a semantic fusion or ranking of answers across sources.

6. QA on the SW: Achievements and research gaps

An overview of related work shows a wide range of approaches that have attempted to support end users in querying and exploring the publicly available SW information. It is not our intention to exhaustively cover all existing approaches, but to look at the state of the art and applications to figure out the capabilities of the different approaches, considering each of the querying dimensions presented in Section 2 (sources, scope, search environment and input), to identify promising directions towards overcoming their limitations and filling the research gaps.

6.1. Sources for QA and their effect on scalability

We have shown through this paper that ontologies are a powerful source to provide semantics and background knowledge about a wide range of domains, providing a new important context for QA systems.

[17] http://challenge.semanticweb.org/

[18] http://www.neofonie.de/index.jsp

- Traditionally, the major drawbacks of intelligent NLIDB systems are that to perform both complex semantic interpretations and achieve high performance, these systems tend to use computationally intensive algorithms for NLP and presuppose large amounts of domain dependent background knowledge and hand-crafted customizations, thus being not easily adaptable or portable to new domains.
- Open QA systems over free text require complicated designs and extensive implementation efforts, due to the high linguistic variability and ambiguity they have to deal with to extract answers from very large open-ended collections of unstructured text. The pitfalls of these systems arise when a correct answer is unlikely to be available in one document but must be assembled by aggregating answers from multiple ones.
- Ontology-specific QA systems, although ontology-independent, are still limited by the single ontology assumption and they have not been evaluated with large-scale datasets.
- Proprietary QA systems, although they scale to open and large scenarios in a potentially unlimited number of domains, cannot be considered as interfaces to the SW, as they use their own encoding of the sources. Nonetheless, they are a good example of open systems that integrate structured and non-structured sources, although, currently they are limited to Wikipedia (Powerset, TrueKnowledge) or a set of annotated documents linked to the KB (START).
- Although not all keyword-based and semantic search interfaces (including facets) scale to multiple sources in the SW, we are starting to see more and more applications that can scale, by accessing search engines (e.g., mash-ups), large collections of datasets (i.e., provided by the billion triple challenge), SPARQL endpoints, or various distributed online repositories (previously indexed). We have also seen an example of semantic search approaches [29] that can retrieve accurate results on the Web. However, this approach is limited by the single-ontology assumption and it is based on the assumption that documents in the Web are annotated. In [29] conjunctive semantic search queries are not formulated yet in NL and logical queries need to be created according to the underlying ontology, thus making the approach inaccessible for the typical Web user. DBpedia has also been used as a source for a query completion component in normal Web queries on the mainstream Yahoo search engine [71]. However, the current implementation is based on a large but single dataset and the results of a large-scale evaluation suggested that the most common queries were not specific enough to be answered by factual data. Thus, factual information may only address a relatively small portion of the user information needs.
- Open Semantic QA approaches, as seen in [31] based on a NL interface to SW repositories and a scalable IR system to annotate and rank the documents in the search space, can in principle scale to the Web and to multiple repositories in the SW in a potentially wide number of domains. However, semantic indexes need to be created offline for both ontologies and documents. Although, also coupled with Watson, its performance with the search engine has not been formally evaluated.

Notwithstanding, we believe that open semantic ontology-based QA systems can potentially fill the gap between closed domain QA over structured sources (NLIDB) and domain independent QA over free text (Web), as an attempt to solve some of the limitations of these two different research areas (see Table 3). Ontology-based QA systems are able to handle a much more expressive and structured search space. Semantic QA systems have proven to be ontology independent (Section 4.1) and even able to perform QA in open domain environments by assembling and aggregating answers from multiple sources (Section 4.3). Finally, the integration of semantic and non-semantic data is an important challenge for future work on ontology-based QA. Current implementations, in particular those based on a limited number of sources, still suffer from the knowledge incompleteness and sparseness problems.

6.2. Scope and tendencies towards open QA approaches

One main dimension over which these approaches can be classified is their scope. On a first level we can distinguish the **closed domain approaches**, whose scope is limited to one (or a set of) a-priori selected domain(s) at a time. As we have seen, ontology-based QA systems, which give meaning to the queries expressed by a user with respect to the domain of the underlying ontology, although portable, their scope is limited to the amount of knowledge encoded in one ontology (they are brittle). As such, they are closer to NLIDB, focused on the exploita-

Table 3
Querying approaches classified according to their intrinsic problems and search criteria

Criteria	Input		Scope			Search environment (research issues)				Sources	
	Expressivity	Reasoning services	Portability	Open Domain	Heterogeneity	Ranking	Disambiguat.	Fusion	Sources on-the-fly	Scale SW	Scale Web
NLIDB	√	√	Ø	Ø	Ø	Ø	√	Ø	Ø	Ø	Ø
QA-Text/ Web	√	Ø	√	√	√	√	√	Ø	√	Ø	√
Ontology-QA	√	√	√	Ø	Ø	+/-	√	Ø	Ø	+/-	Ø
Proprietary QA	√	√	√	√	Ø	√	√	Ø	Ø	Ø	+/-
Keyword-search	+/-	Ø	√	√	√	√	+/-	Ø	√	√	+/-
Mash-ups	Ø	Ø	√	√	√	+/-	Ø	√	√	√	Ø
Facets	√	Ø	√	√	√	√	Ø	Ø	Ø	√	Ø
Semantic open QA	√	Ø	√	√	√	√	√	√	+/-	√	+/-

tions of unambiguous structured data in closed-domain scenarios to retrieve precise answers to questions, than to QA over a document collection or free text. While these approaches have proved to work well when a pre-defined domain ontology is used to provide an homogenous encoding of the data, none of them can handle complex questions by combining domain specific information typically expressed in different heterogeneous sources.

On a second level, and enhancing the scope embraced by closed domain models, we can distinguish those **approaches restricted to their own semantic resources**. While successful NL search interfaces to structured knowledge in an open domain scenario exist (popular examples are Powerset or TrueKnowledge), they are restricted to the use of their own semi-automatically built and comprehensive factual knowledge bases. This is the most expensive scenario as they are typically based on data that are by and large manually coded and homogeneous.

On a third level, we can highlight the latest **open semantic search approaches**. These systems are not limited by closed-domain scenarios, neither by their own resources, but provide a much wider scope, attempting to cover and reuse the majority of publicly available semantic knowledge. We have seen examples of these different approaches: a) using Linked Data sources, i.e., DBpedia, for a query completion component on the Yahoo search engine, b) keyword-based query interfaces to data sources available in the billion triple challenge datasets and live SPARQL endpoints, c) mash-ups able to aggregate heterogeneous data obtained from the search engine Sindice from a given keyword, d) Open Linked Data facets, which allow the user to filter objects according to properties or range of values, and e) NL QA system over multiple heterogeneous semantic repositories, including large Linked Data sources (i.e. DBpedia) and (with some decrease in performance) the search engine Watson.

We can see that there is a continuous tendency to move towards applications that take advantage of the vast amount of heterogeneous semantic data and get free of the burden of engineering their own semantic data. Hence, as predicted by [81], we are heading into a new generation of semantic systems [24], able to explore the SW as a whole and handle the scalability, heterogeneity and openness issues posed by this new challenging environment.

As such, the next key step towards the realization of QA on the SW is to move beyond domain specific semantic QA to robust open domain semantic QA over structured and distributed semantic data. In this direction the PowerAqua system provides a single NL access approach for all the diverse online resources, stored in multiple collections, opening the possibility of searching and combining answers from all the resources together. Nonetheless, as seen in [63], it is often the case that queries can only be solved by composing information derived from multiple and autonomous information sources, hence, portability alone is not enough and openness is required. QA systems able to draw precise, focused answers by locating and integrating information, which can be distributed across heterogeneous and distributed semantic sources, are required to go beyond the state of the art in interfaces to query the SW.

6.3. Traditional intrinsic problems derived from the search environment

A new layer of complexity arises when moving from a classic KB system to an open and dynamic search environment. If an application wishes to use data from multiple sources the integration effort is non-trivial.

While the latest open Linked Data and semantic search applications shown in Section 5.3 present a much wider scope, scaling to the large amounts of available semantic data, they perform a shallow exploitation of this information: 1) they do not perform semantic disambiguation, but need users to select among possible query interpretations, 2) they do not generally provide knowledge fusion and ranking mechanisms to improve the accuracy of the information retrieved, and 3) they do not discover mappings between data sources on the fly, but need to pre-compute them beforehand.

Automatic disambiguation (point 1) can only be performed if the user query is expressive enough to grasp the conceptualizations and content meanings involved in the query. In other words, the context of the query is used to choose the correct interpretation. If the query is not expressive enough, the only alternative is to call the user to disambiguate, or to rank the different meanings based on the popularity of the answers.

Although ontology-based QA can use the context of the query to disambiguate the user query, it still faces difficulties to scale up to large-scale and heterogeneous environments. The complexity arises because of its "openness", as argued in [80], QA systems in restricted domains can attack the answer-retrieval problem by means of an internal unambiguous knowledge representation, however, in open-domain scenarios, or when using open-domain ontologies, as is the case of DBpedia or WordNet that map words to concepts, systems face the problem of polysemous words, which are usually unambiguous in restricted domains. At the same time, open-domain QA can benefit from the size of the corpus: as the size increases it becomes more likely that the answer to a specific question can be found without requiring a complex language model. As such, in a large-scale open scenario the complexity of the tools will be a function of their ability to make sense of the heterogeneity of the data to perform a deep exploitation beyond simple lookup and mash-up services. Moreover, ranking techniques are crucial to scale to large-scale sources or multiple sources.

With regards to fusion (point 2) only mash-ups and open ontology-based QA systems aggregate answers across sources. However, so far, mash-ups do not attempt to disambiguate between the different interpretations of a user keyword.

With regards to on the fly mappings (point 2), most SW systems analyzed here perform mappings on the fly given a user task, and some of them are able to select the relevant sources on the fly. There are three different mechanisms which are employed: (1) through search engines (mash-ups, semantic search, open ontology-based QA); (2) by accessing various distributed online SPARQL end-points providing full text search capabilities (semantic search, facets); (3) by indexing multiple online repositories (open ontology-based QA, semantic search). State of the art open ontology-based QA and semantic search systems perform better by indexing multiple online repositories for its own purposes. When a search engine such as Watson, which provides enough functionality (API) to query and perform a deep analysis of the sources, is used the performance is just acceptable from a research point of view demo [62]. More work is needed to achieve real time performance –beyond prototypes, for ontology-based QA to directly catch and query the relevant sources from a search engine that crawls and indexes the semantic sources.

In Table 3 we compare how the different approaches to query the SW, tackle these traditional intrinsic problems derived from the openness of the search environment (automatic disambiguation of user needs, ranking, portability, heterogeneity and fusion across sources).

6.4. Input and higher expressivity

Finally, the expressivity of the user query is defined by the input the system is able to understand. As shown in Table 3, keyword-based systems lack the expressivity to precisely describe the user's intent, as a result ranking can at best put the query intentions of the majority on top. Most approaches look at expressivity at the level of relationships (factoids), however, different systems provide different support for complex queries, from including reasoning services to understand comparisons, quantifications and negations, to the most complex systems (out of the scope of this review) that go beyond factoids and are able to understand anaphora resolution and dialogs [4]. Ontologies are a powerful tool to provide semantics, and in particular, they can be used to move beyond single facts to enable answers built from multiple sources. However, regarding the input, ontologies have limited capability to reason about temporal and spatial queries and do not typically store time dependent information. Hence, there is a serious research challenge in determining how to handle temporal data and causality across ontologies. In a search system for the open SW we cannot expect complex reasoning over very expressive ontologies, because this requires detailed knowledge of ontology structure. Complex ontology-dependent reasoning is substituted by the ability to deal with and find connections across large amounts of heterogeneous data.

7. Directions ahead

Despite all efforts semantic search still suffers from the knowledge incompleteness problem, together with the cost of building and maintaining rich semantic sources and the lack of ranking algorithms to cope with large-scale information sources [30]. Due to all this, semantic search cannot yet compete with major search engines, like Google, Yahoo or Microsoft Bing[19].

Nonetheless, through efforts such as the Linked Open Data initiative, the Web of Data is becoming a reality, growing and covering a broader range of topics, and it is likely that soon we will have so much data that the core issues would not be only related to sparseness and brittleness, as to scalability and robustness. Novel approaches that can help the typical Web user to access the open, distributed, heterogeneous character of the SW and Linked Data are needed to support an effective use of this resource.

Scalability is a major open issue and study presented in [56] about the potential size of the SW reveals that the SW mirrors the growth of the Web in its early stages. Therefore, semantic systems should be able to support large-scale data sources both in terms of ontology size and the number of them (as of September 2011 the Linked Data Cloud contained more than 19 billion triples).

While semantic search technologies have been proven to work well in specific domains still have to confront many challenges to scale up to the Web in its entirety. The latest approaches to exploit the massive amount of distributed SW data represent a considerable advance with respect to previous systems, which restrict their scope to a fraction of the publicly available SW content or rely on their own semantic resources. These approaches are ultimately directed by the potential capabilities of the SW to provide accurate responses to NL user queries, but are NL QA approaches fit for the SW?

In this scenario, QA over semantic data distributed across multiple sources has been introduced as a new paradigm, which integrates ideas of traditional QA research into scalable SW tools. In our view, there is great potential for open QA approaches in the SW. As shown in Table 3 semantic open QA has tackled more problems than other methods for many of the analyzed criteria. In an attempt to overcome the limitations of search approaches, that restrict their scope to homogenous or domain-specific content, or perform a shallow exploitation of it, current QA systems have developed syntactic, semantic and contextual information processing mechanisms that allow a deep exploitation of the semantic information space.

As such, we believe that open semantic QA is a promising research area that goes beyond the state of the art in user-friendly interfaces to support users in querying and exploring the heterogeneous SW content. In particular:

- To bridge the gap between the end-user and the real SW by providing a NL QA interface that can scale up to the Web of Data.
- To take advantage of the structured information distributed on the SW to retrieve aggregate answers to factual queries that extend beyond the coverage of single datasets and are built across multiple ontological statements obtained from different sources. Consequently, smoothing the habitability and brittleness problems intrinsic to closed domain KB systems.

The ultimate goal for a NL QA system in the SW is to answers queries by locating and combining information, which can be massively distributed across heterogeneous semantic resources, without imposing any pre-selection or pre-construction of semantic knowledge, but rather locating and exploring the increasing number of multiple, heterogeneous sources currently available on the Web.

Performance and scalability issues still remain open. Balancing the complexity of the querying process in an open-domain scenario (i.e., the ability to handle complex questions requiring making deductions on open-domain knowledge, capture the interpretation of domain-specific adjectives, e.g., “big”, “small”, and in consequence superlatives, e.g., “largest”, “smallest” [16], or combining domain specific information typically expressed in different sources)

[19] Google: http://www.google.com, Yahoo! Search: http://www.yahoo.com, Bing: http://www.bing.com

and the amount of semantic data is still an open problem. The major challenge is, in our opinion, the combination of scale with the considerably heterogeneity and noise intrinsic to the SW. Moreover, information on the SW originates from a large variety of sources and exhibits differences in granularity and quality, and therefore, as the data is not centrally managed or produced in a controlled environment, quality and trust become an issue. Publishing errors and inconsistencies arise naturally in an open environment like the Web [84]. Thus, imperfections (gaps in coverage, redundant data with multiple identifiers for the same resource, conflicting data, undefined classes, properties without a formal schema description, invalid datatypes, etc.) can be seen as an inherent property of the Web of Data. As such, the strength of the SW will be more a by-product of its size than its absolute quality.

Thus, in factual QA systems over distributed semantic data the lack of very complex reasoning is substituted by the ability to deal and find connections in large amounts of heterogeneous data and to provide coherent answers within a specific context or task. As a consequence, exploiting the SW is by and large about discovering interesting connections between items. We believe that in those large scale semantic systems, intelligence becomes a side effect of a system's ability to operate with large amounts of data from heterogeneous sources in a meaningful way rather than being primarily defined by their reasoning ability to carry out complex tasks. In any case this is unlikely to provide a major limitation given that, most of the large datasets published in Linked Data are light-weight.

Furthermore, besides scaling up to the SW in its entirety to reach the full potential of the SW, we still have to bridge the gap between the semantic data and unstructured textual information available on the Web. We believe, that as the number of annotated sites increases, the answers to a question extracted in the form of lists of entities from the SW, can be used as a valuable resource for discovering Web content that is related to the answers given as ontological entities. Ultimately, complementing the structured answers from the SW with Web pages will enhance the expressivity and performance of traditional search engines with semantic information.

Acknowledgments

Research funded under the Smart-Products European project (EC-231204).

References

[1] B. Aleman-Meza, F. Hakimpour, I.B. Arpinar and A.P. Sheth, SwetoDblp ontology of computer science publications, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* **5**(3) (2007), 151–155. Elsevier Science Publishers B.V. Amsterdam.

[2] I. Androutsopoulos, G.D. Ritchie and P. Thanisch, MASQUE/SQL – An efficient and portable natural language query interface for relational databases, in: *Proc. of the 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Edinburgh*, Gordon and Breach Publisher Inc., 1993, pp. 327–330.

[3] I. Androutsopoulos, G.D. Ritchie and P. Thanisch, Natural language interfaces to databases – An introduction, *Natural Language Engineering* **1**(1) (1995), 29–81. Cambridge University Press.

[4] R. Basili, D. De Cao and C. Giannone, Ontological modeling for interactive question answering, in: *Proc. of the OTM Confederated International Conference on "On the Move to Meaningful Internet Systems* (1)*"*, Z. Tari, ed., Springer-Verlag, 2007, pp. 544–553.

[5] R. Basili, D.H. Hansen, P. Paggio, M.T. Pazienza and F.M. Zanzotto, Ontological resources and question answering, *Workshop on Pragmatics of Question Answering, Held Jointly with HLT-NAACL 2004*, Boston, Massachusetts, 2004.

[6] R.A. Baeza and P. Raghavan, Next generation Web search, in: *Search Computing*, S. Ceri and M. Brambilla, eds, LNCS, Vol. 5950, Springer, 2010, pp. 11–23.

[7] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, *Scientific American* **284**(5) (2001), 33–43.

[8] A. Bernstein, E. Kauffmann, C. Kaiser and C. Kiefer, Ginseng: A guided input natural language search engine, in: *Proc. of the 15th Workshop on Information Technologies and Systems (WITS 2005)*, MV-Wissenschaft, Münster, 2006, pp. 45–50.

[9] C. Bizer, T. Heath and T. Berners-Lee, Linked data – The story so far, *International Journal on Semantic Web and Information Systems* **5**(3) (2009), 1–22. Elsevier.

[10] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak and S. Hellman, DBPedia. A crystallization point for the Web of data, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* **7**(3) (2009), 154–165. Elsevier.

[11] P. Bouquet, L. Serafini and S. Zanobini, Semantic coordination: A new approach and an application, in: *Proc. of 2nd International Semantic Web Conference*, K. Sycara, and J. Mylopoulos, eds, LNCS, Vol. 2870, Springer, Sanibel Island, Florida, 2003, pp. 130–145.

[12] R.D. Burke, K.J. Hammond and V. Kulyukin, Question Answering from frequently-asked question files: Experiences with the FAQ finder system, in: *Proc. of the World Wide Web Internet and Web Information Systems* **18**(TR-97-05) (1997), 57–66. Department of Computer Science, University of Chicago.

[13] P. Castells, M. Fernández and D. Vallet, An adaptation of the vector-space model for ontology-based information retrieval, *IEEE Transactions on Knowledge and Data Engineering* **19**(2) (2007), 261–272. IEEE Educational Activities Department.

[14] G. Cheng, W. Ge and Y. Qu, Falcons: Searching and browsing entities on the Semantic Web, in: *Proc. of the 17tth International Conference on World Wide Web*, M. Wei-Ying,

A. Tomkins and X. Zhang, eds, ACM, Beijing, China, 2008, pp. 1101–1102.

[15] P. Cimiano, P. Haase and J. Heizmann, Porting natural language interfaces between domains – An experimental user study with the ORAKEL system, in: *Proc. of the International Conference on Intelligent User Interfaces*, D.N. Chin, M.X. Zhou, T.S. Lau and A.R. Puerta, eds, ACM, Gran Canaria, Spain, 2007, pp. 180–189.

[16] P. Cimiano and M. Minock, Natural language interfaces: what is the problem? – A data-driven quantitative analysis, in: *Proc. of the 14th International Conference on Applications of Natural Language to Information Systems*, H. Horacek, E. Metais, R. Munoz and M. Wolska, eds, Springer, Saarland, Germany, 2009, pp. 192–206.

[17] W.W. Cohen, P. Ravikumar and S.E. Fienberg, A comparison of string distance metrics for name-matching tasks, *IJCAI Workshop on Information Integration on the Web (IIWeb-03)*, Acapulco, Mexico, 2003, pp. 73–78.

[18] A. Copestake and K.S. Jones, Natural language interfaces to databases, *Knowledge Engineering Review* **5**(4) (1990), 225–249.

[19] Croft, User-specified domain knowledge for document retrieval, in: *Proc. of the 9th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR 1986)*, F. Rabitti, ed., ACM, Pisa, Italy, 1986, pp. 201–206.

[20] H. Cunningham, D. Maynard, K. Bontcheva and V. Tablan, GATE: A framework and graphical development environment for robust NLP tools and applications, in: *Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics* **54** (2002), 168–175. Association for Computational Linguistics.

[21] D. Damljanovic, M. Agatonovic and H. Cunningham, Natural Language interface to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction, in: *Proc. of the European Semantic Web Conference, Heraklion, Greece*, L. Aroyo, G. Antoniou, E. Hyvönen, A. ten Teije, H. Stuckenschmidt, L. Cabral and T. Tudorache, eds, Springer Verlag, 2010.

[22] D. Damljanovic, V. Tablan and K. Bontcheva, A text-based query interface to OWL ontologies, in: *Proc. of the 6th Language Resources and Evaluation Conference (LREC)*, European Language Resources Association (ELRA), Marrakech, Morocco, 2008, pp. 205–212.

[23] M. D'Aquin, C. Baldassarre, L. Gridinoc, S. Angeletou, M. Sabou and E. Motta, Characterizing knowledge on the semantic web with Watson, in: *Proc. of 5th EON Workshop at International Semantic Web Conference*, Citeseer, Busan, Korea, 2007, pp. 1–10.

[24] M. D'Aquin, E. Motta, M. Sabou, S. Angeletou, L. Gridinoc, V. Lopez and D. Guidi, Towards a new generation of Semantic Web applications, *IEEE Intelligent Systems* **23**(3) (2008), 20–28.

[25] A.N. De Roeck, C.J., Fox, B.G.T. Lowden, R. Turner and B. Walls, A natural language system based on formal semantics, in: *Proc. of the International Conference on Current Issues in Computational Linguistics, Penang, Malaysia*, 1991.

[26] L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng and P. Kolari, Finding and ranking knowledge on the Semantic Web, in: *Proc. of International Semantic Web Conference*, Y. Gil, E. Motta, V.R. Benjamins and M. Musen, eds, LNCS, Vol. 3729, Springer, Galway, Ireland, 2005, pp. 156–170.

[27] M. Dittenbach, D. Merkl and H. Berger, A natural language query interface for tourism information, in: *Proc. of the 10th International Conference on Information Technologies in Tourism (ENTER-03)*, Springer Verlag, Helsinki, Finland, 2003, pp. 152–162.

[28] O. Erling and I. Mikhailov, Faceted views over large-scale linked data, in: Proc. of the Linked Data on the Web Workshop at the World Wide Web Conference (LDOW2009), Madrid, Spain, 2009.

[29] B. Fazzinga and T. Lukasiewicz, Semantic search on the Web, *SemanticWeb Journal – Interoperability, Usability, Applicability* **1**(1–2) (2010), 1–7. IOS Press.

[30] M. Fernandez, I. Cantandor, V. Lopez, D. Vallet, P. Castells, and E. Motta, Semantically enhanced information retrieval: An ontology-based approach, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web. Special Issue on Semantic Search,* 2010. In Press.

[31] M. Fernandez, V. Lopez, M. Sabou, V. Uren, D. Vallet, E. Motta and P. Castells, Semantic search meets the Web, in: *Proc. of the IEEE International conference on Semantic Computing*, IEEE Computer Society, Santa Clara, California, 2008, pp. 253–260.

[32] O. Fernandez, R. Izquierdo, S. Ferrandez and J.L. Vicedo, Addressing Ontology-based question answering with collections of user queries, *Information Processing and Management* **45**(2) (2009), 175–188. Elsevier.

[33] P. Forner, D. Giampiccolo, B. Magnini, A. Peñas, A. Rodrigo and R. Sutcliffe, Evaluating multilingual question answering systems at CLEF, in: *Proc. of the Conference on International Language Resources and Evaluation (LREC), Malta*, European Language Resources Association, 2010.

[34] A. Frank, K. Hans-Ulrich, X. Feiyu, U. Hans, C. Berthold, J. Brigitte and S. Ulrich, Question answering from structured knowledge sources, *Journal of Applied Logic, Special Issue on Questions and Answers: Theoretical and Applied Perspectives* **5**(1) (2006), 20–48. Elsevier.

[35] B.F. Green, A.K. Wolf, C. Chomsky and K. Laughery, BASEBALL: An automatic question answerer, *Proceedings Western Joint Computer Conference* **19** (1961), 207–216. McGraw-Hill.

[36] T.R. Gruber, A translation approach to portable ontology specifications, *Knowledge Acquisition* **5**(2) (1993), 199–220. Elsevier.

[37] N. Guarino, C. Masolo and G. Vetere, OntoSeek: Content-based access to the Web, *IEEE Intelligent Systems* **14**(3) (1999), 70–80.

[38] C. Gueret, P. Groth and S. Schlobach, eRDF: Live discovery for the Web of data, *Submission for the Billion Triple Challenge at International Semantic Web Conference*, LNCS, Vol. 5823, Springer, Chantilly, VA, USA, 2009.

[39] R.V. Guha, R. McCool and E. Miller, Semantic search, *Proc. of the 12th International World Wide Web Conference (WWW 2003)* **36**(1) (2003), 700–709. Budapest. ACM Press.

[40] I. Gurevych, D. Bernhard, K. Ignatova and C. Toprak, Educational question answering based on social media content, in: *Proc. of the 14th International Conference on Artificial Intelligence in Education*, **200** (2009), 133–140. IOS Press.

[41] C. Hallett, D. Scott and R. Power, Composing questions through conceptual authoring, *Computational Linguistics* **33**(1) (2007), 105–133. MIT Press.

[42] S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus and P. Morarescu, Falcon – Boosting knowledge for answer engines, in: *Proc. of the 9th Text Retrieval Conference (Trec-9)*, 2000, pp. 479–488.

[43] J. Hendler, Web 3.0: The dawn of semantic search, *IEEE Computer* **43**(1) (2010), 77–80. IEEE Computer Society Press.

[44] M. Hildebrand, J. Ossenbruggen and L. van Hardman, An analysis of search-based user interaction on the Semantic Web, *Information Systems Journal, INS-E0706*, 2007, pp. 1386–3681. CWI, Amsterdam, Holland.

[45] L. Hirschman and R. Gaizauskas, Natural language question answering: The view from here, *Natural Language Engineering, Special Issue on Question Answering* **7**(4) (2001), 275–300. Cambridge University Press.

[46] E.H. Hovy, L. Gerber, U. Hermjakob, M. Junk and C.-Y. Lin, Question answering in webclopedia, in: *Proc. of the Ninth Text Retrieval Conference (TREC-9). NIST*, 2000, pp. 655–664.

[47] A. Hunter, Natural Language database interfaces, *Knowledge Management*, 2000, www.cs.ucl.ac.uk/staff/a.hunter/tradepress/nldb.html.

[48] E. Hyvonen, E. Makela, M. Salminen, A. Valo, K. Viljanen, S. Saarela, M. Junnila and S. Kettula, Museum Finland – Finnish museums on the Semantic Web, *Journal of Web Semantics: Science Services and Agents on the World Wide Web. Selected Papers from the International Semantic Web Conference* 2004, **3**(2) (2005), 224–241.

[49] B. Katz, S. Felshin, D. Yuret, A. Ibrahim, J. Lin, G. Marton, A.J. McFarland and B. Temelkuran, Omnibase: Uniform access to heterogeneous data for question answering, in: *Proc. of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB)*, Stockholm, Sweden, 2002, pp. 230–234.

[50] E. Kaufmann, Talking to the Semantic Web – Natural language query interfaces for casual end-users, PhD thesis, University of Zurich, Switzerland, 2009.

[51] E. Kaufmann and A. Bernstein, How useful are natural language interfaces to the Semantic Web for casual end-users?, in: *Proc. of the 6th International Semantic Web Conference*, K. Aberer, K.S. Choi, N. Noy, D. Allemang, K.I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber and P. Cudré-Mauroux, eds, LNCS, Vol. 4825, Springer, Busan, Korea, 2007, pp. 281–294.

[52] E. Kauffmann, A. Bernstein and L. Fischer, NLP-Reduce: A "naïve" but domain-independent natural language interface for querying ontologies, in: *Proc. of the 4th European Semantic Web Conference*, E. Franconi, M. Kifer and W. May, eds, Springer Verlag, Innsbruck, 2007, pp. 1–2.

[53] E. Kauffmann, A. Bernstein and R. Zumstein, Querix: A natural language interface to query ontologies based on clarification dialogs, in: *Proc. of the 5th International Semantic Web Conference*, I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold and L. Aroyo, eds, LNCS, Vol. 4273, Springer, Athens, USA, 2006, pp. 980–981.

[54] D. Klein and C.D. Manning, Fast exact inference with a factored model for natural language parsing, *Advances in Neural Information Processing Systems* **15** (2002), 3–10. Citeseer.

[55] C. Kwok, O. Etzioni and D. Weld, Scaling question answering to the Web, in: *Proc. of the 10th International Conference on World Wide Web*, ACM, Hong Kong, China, 2001, pp. 150–161.

[56] J. Lee and R. Goodwin, The semantic webscape: A view of the Semantic Web, in: *Special interest tracks and posters of the 14th International Conference on World Wide Web, Chiba, Japan*, A. Ellis and T. Hagino, eds, ACM, 2005.

[57] Y. Lei, V. Uren and E. Motta, SemSearch: A search engine for the Semantic Web, in: *Proc. of the 15th International Conference of Knowledge Engineering and Knowledge Management, EKAW*, S. Staab and V. Svátek, eds, Springer Verlag, Podebrady, Czech Republic, 2006, pp. 238–245.

[58] A.Y. Levy, D. Srivastava and T. Kirk, Data model and query evaluation in global information systems, *Journal of Intelligence Information Systems, JIIS* **5**(2) (1995), 121–143. Springer.

[59] S. Linckels and C. Meinel, A simple solution for an intelligent librarian system, in: *Proc. of the IADIS International Conference of Applied Computing*, 2005, pp. 495–503.

[60] S. Linckels and C. Meinel, Resolving ambiguities in the semantic interpretation of natural language questions, in: *Proc. of the Intelligence Data Engineering and Automatic Learning, IDEAL*, LNCS, Vol. 4224, 2006, pp. 612–619.

[61] K.C. Litkowski, Syntactic clues and lexical resources in question-answering, *Information Technology: The Ninth Text REtrieval Conference (TREC-9)*, NIST Special Publication 500-249, 2001.

[62] V. Lopez, M. Fernandez, E. Motta and N. Stieler, PowerAqua: Supporting users in querying and exploring the Semantic Web, To appear in the Journal of Semantic Web – Interoperability, Usability, Applicability, IOS Press, 2011.

[63] V. Lopez, A. Nikolov, M. Fernandez, M. Sabou, V. Uren, and E. Motta, Merging and ranking answers in the Semantic Web: The wisdom of crowds, in: *Proc. of the Asian Semantic Web Conference*, Y. Yong and Y. Ding, eds, Springer, Berlin Heidelberg, Shanghai, China, 2009, pp. 135–152.

[64] V. Lopez, A. Nikolov, M. Sabou, V. Uren, E. Motta and M. d'Aquin, Scaling up question-answering to linked data, in: *Proc. of the 17th Knowledge Engineering and Knowledge Management by the Masses*, P. Cimiano and S. Pinto, eds, Springer, Berlin Heidelberg, Lisbon, Portugal, 2010, pp. 193–210.

[65] V. Lopez, M. Sabou and E. Motta, PowerMap: Mapping the real Semantic Web on the fly, in: *Proc. of the 5th International Semantic Web Conference*, I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold and L. Aroyo, eds, LNCS, Vol. 4273, Springer, Athens, USA, 2006, pp. 5–9.

[66] V. Lopez, M. Sabou, V. Uren and E. Motta, Cross-Ontology question answering on the Semantic Web – An initial evaluation, in: *Proc. of the Knowledge Capture Conference*, Y. Gil and N. Noy, eds, ACM, California, USA, 2009, pp. 17–24.

[67] V. Lopez, V. Uren, E. Motta and M. Pasin, AquaLog: An ontology-driven question answering system for organizational semantic intranets, *Journal of Web Semantics: Science Service and Agents on the World Wide Web* **5**(2) (2007), 72–105.

[68] P. Martin, D.E. Appelt, B.J. Grosz and F.C.N. Pereira, TEAM: An experimental transportable natural-language interface, *IEEE Database Engineering* **8**(3) (1985), 10–22.

[69] R. McCool, A.J. Cowell and D.A. Thurman, End-user evaluations of Semantic Web technologies, Workshop on End User Semantic Web Interaction, in: *Proc. of the International Semantic Web Conference*, LNCS, Springer, 2005.

[70] D. McGuinness, Question answering on the Semantic Web, *IEEE Intelligent Systems* **19**(1) (2004), 82–85.

[71] E. Meij, P. Mika and H. Zaragoza, Investigating the demand side of semantic search through query log analysis, in: *Proc. of the Workshop on Semantic Search at the 18th International World Wide Web Conference (WWW 2009)*, Madrid, Spain, 2009.

[72] E. Mena, V. Kashyap, A. Sheth and A. Illarramendi, OBSERVER: An approach for query processing in global information systems based on interoperation across preexisting ontologies, *Distributed and Parallel Databases* **8**(2) (2000), 223–271.

[73] P. Mika, Flink: SemanticWeb technology for the extraction and analysis of social networks, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* **3**(2) (2005), 211–223. Elsevier.

[74] M. Minock, C-Phrase: A system for building robust natural language interfaces to databases, *Journal of Data Engineering (DKE)* **69**(3) (2010), 290–302. Elsevier.

[75] M. Minock, P. Olofsson and A. Naslund, Towards building robust natural language interfaces to databases, in: *Proc. of the 13th international conference on Natural Language and Information Systems, London, UK*, 2008.

[76] S. Mithun, L. Kosseim and V. Haarslev, Resolving quantifier and number restriction to question OWL ontologies, in: *Proc. of The First International Workshop on Question Answering at the International Conference on Semantics, Knowledge and Grid, Xi'an, China*, 2007.

[77] D. Moldovan, S. Harabagiu, et al., LCC tools for question answering, in: *Proc. of the Text Retrieval Conference TREC-11*, 2002.

[78] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Goodrum, R., Girju and V. Rus, LASSO: A tool for surfing the answer net, in: *Proc. of the Text Retrieval Conference (TREC-8)*, 1999.

[79] D. Moldovan, M. Pasca, S. Harabagiu and M. Surdeanu, Performance issues and error analysis in an open-domain question answering system, *ACM Trans. Information Systems* **21**(2) (2003), 133–154.

[80] D. Mollá and J.L. Vicedo, Question answering in restricted domains: An overview, *Computational Linguistics* **33**(1) (2007), 41–61. MIT Press.

[81] E. Motta and M. Sabou, Language technologies and the evolution of the semantic web, in: *Proc. of the Fifth International Conference on Language Resources and Evaluation (LREC)*, ELRA, Genoa, Italy, 2006.

[82] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhor and G. Tummarello, Sindice.com: A document-oriented lookup index for open linked data, *International Journal of Metadata, Semantics and Ontologies* **3**(1) (2008), 37–52. Inderscience Publishers.

[83] M. Pasca, Open-domain question answering from large text collections, *CSLI Studies in Computational Linguistics* **29**(4) (2003), 46. CSLI Publications.

[84] A. Polleres, A. Hogan, A. Harth and S. Decker, Can we ever catch up with the Web?, *Journal of Semantic Web – Interoperability, Usability, Applicability* **1**(1–2) (2010), 45–52. IOS Press.

[85] A.M. Popescu, O. Etzioni and H.A. Kautz, Towards a theory of natural language interfaces to databases, in: *Proc. of the International Conference on Intelligent User Interfaces*, ACM Press, 2003, pp. 149–157.

[86] M.C. Schraefel, N. Shadbolt, N. Gibbins, H. Glaser and S. Harris, CS AKTive space: Representing computer science in the Semantic Web, in: *Proc. of the International World Wide Web Conference*, S. Feldman, M. Uretsky, M. Najork, and C. Wills, eds, ACM Press, New York, USA, 2004, pp. 384–392.

[87] K. Srihari, W. Li and X. Li, Information extraction supported question- answering, in: *Advances in Open – Domain Question Answering*. Kluwer Academic Publishers, 2004.

[88] V. Tablan, D. Damljanovic and K. Bontcheva, A natural language query interface to structured information, in: *Proc. of the 5th European Semantic Web Conference*, S. Bechhofer, M. Hauswirth, J. Hoffmann and M. Koubarakis, eds, Springer Verlag, Tenerife, Spain, 2008, pp. 1–15.

[89] L.R. Tang and R.J. Mooney, Using multiple clause constructors in inductive logic programming for semantic parsing, in: *Proc. of the 12th European Conference on Machine Learning (ECML-2001)*, L. Raedt and P. Flanch, eds, Springer, Germany, 2001, pp. 466–477.

[90] S. Tartir and I.B. Arpinar, Question answering in linked data for scientific exploration, in: *The 2nd Annual Web Science Conference*, ACM, 2010.

[91] C.W. Thompson, P. Pazandak and H.R. Tennant, Talk to your Semantic Web, *IEEE Internet Computing* **9**(6) (2005), 75–78. IEEE Computer Society.

[92] T. Tran, P. Cimiano, S. Rudolph and R. Studer, Ontology-based interpretation of keywords for semantic search, in: *Proc. of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference*, K. Aberer, K.S. Choi, N. Noy, D. Allemang, K.I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber and P. Cudré-Mauroux, eds, LNCS, Vol. 4825, Springer, Busan, Korea, 2007, pp. 523–536.

[93] G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru and S. Decker, Sig.ma: Live views on the Web of dta, in: Rappa, M., Jones, P., Freire, J., and Chakrabarti, S., eds, *Proc. World Wide Web Conference (WWW-2010)*, **8**(4) (2010), 1301–1304. Elsevier B.V., Raleigh, USA.

[94] V. Uren, Y. Lei, V. Lopez, H. Liu, E. Motta and M. Giordanino, The usability of semantic search tools: A review, *Knowledge Engineering Review* **22**(4) (2007), 361–377.

[95] V. Uren, M. Sabou, E. Motta, M. Fernandez, V. Lopez and Y. Lei, Reflections on five years of evaluating semantic search systems, *International Journal of Metadata, Semantics and Ontologies* **5**(2) (2010), 87–98. Inderscience Publishers.

[96] C. Wang, M. Xiong, Q. Zhou and Y. Yu, PANTO: A portable natural language interface to ontologies, in: *Proc. of the 4th European Semantic Web Conference*, E. Franconi, M. Kifer and W. May, eds, Springer Verlag, Innsbruck, Austria, 2007, pp. 473–487.

[97] H. Wang, T. Tran, P. Haase, T. Penin, Q. Liu, L. Fu and Y. Yu, SearchWebDB: Searching the Billion Triples!, *Billion Triple Challenge 2008 at the International Semantic Web Conference, Karlsruhe, Germany*, 2008.

[98] W. Woods, Progress in natural language understanding – An application to lunar geology, *Proc. of the American Federation of Information Processing Societies (AFIPS)* **42** (1973), 441–450. AFIPS Press.

[99] M. Wu, X. Zheng, M. Duan, T. Liu and T. Strzalkowski, Question answering by pattern matching, web-proofing, semantic form proofing, *NIST Special Publication: The Twelfth Text REtrieval Conference (TREC)*, 2003, pp. 500–255.

[100] G. Zenz, X. Zhou, E. Minack, W. Siberski and W. Nejdl, From keywords to semantic queries—Incremental query construction on the Semantic Web, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* **7**(3) (2009), 166–176. Elsevier.